National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

NL-339 (r. 88/04)

Canada

The University of Alberta

A Simulator for Robots with Flexible Links

by

David J. Mackay

A thesis
submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering

Edmonton, Alberta

Fall, 1988

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-45702-3

The University of Alberta

Release Form

Name of Author:	David J. Mackay
Title of Thesis:	A Simulator for Robots with Flexible Links
Degree:	Master of Science
Year this degree granted:	Fall 1988

Permission is hereby granted to **The University of Alberta Library** to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly, or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive tracts from it may be printed or otherwise reproduced without the author's written consent.

.....D.J.Mackay.....

(Student's signature)

...13...Ross Glen Way S.E...

....Medicine Hat.....

....Alberta....T1B 1N9.....

(Student's permanent address)

Date : ...Sept 8, 88.

The University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled

A Simulator for Robots with Flexible Links

submitted by

David J. Mackay

in partial fulfilment of the requirements for the degree of

Master of Science.

Rajan W. Toogood
(Supervisor)
V. J. Gourishankar
A. Lipsett
S. Kennedy

Date : 29 April '88

Abstract

A simulator, called FLX, for flexible link manipulators with revolute joints has been developed. The derivation of the flexible link dynamics uses a 4×4 homogeneous transformation matrix Langrangian formulation. Link flexbility is included using the method of assumed modes.

The kinematics and dynamics of rigid link manipulators, in particular, the recursive 4×4 matrix Lagrangian formulation of the dynamics, are reviewed. And the extension of the 4×4 matrix Lagrangian formulation to include flexibility is presented.

Attempts to verify the correctness of solutions produced by the simulator are presented. The simulator is shown to produce solutions that are conservative in the absence of external forcing. And, when modeling simple geometries undergoing two dimensional motion, FLX is shown to produce solutions that agree with those produced using a scalar Lagrangian formulation.

The computational expense of the simulation is discussed and shown to be greatly reduced if the computer code for evaluating the equations of motion is developed symbolically.

Acknowledgements

I wish to thank my supervisor, Dr. Roger Toogood, for his guidance and advice throughout the course of this work.

I also wish to express my gratitude for the financial support provided me during the course of this research by the Natural Sciences and Engineering Research Council (NSERC), the University of Alberta Department of Mechanical Engineering, and the Defence Research Establishment Suffield (DRES).

Contents

Abstract iv

Acknowledgements v

Chapter 1

Introduction 1

1.1 Rigid Link Manipulator Kinematics 3

1.2 Rigid Link Manipulator Dynamics 4

1.3 Symbolic Generation of Dynamics Equations 9

1.4 Outline of This Work 10

Chapter 2

Derivation of the Equations of Motion 12

2.1 Kinematics 13

2.2 Dynamics 21

2.2.1 Kinetic Energy 21

2.2.2 Potential Energy 24

2.3 Assembly of the Equations of Motion 29

2.3.1 Inertia Coefficients 29

2.3.2 Remaining Dynamics Terms 33

Chapter 3

Model Implementation and Verification 36

3.1 Overview of FLX 36

3.2 Limitations and assumptions 38

3.3 Model verification 43

3.3.1 Single rigid link compound pendulum 45

3.3.2 Double rigid link compound pendulum 50

3.3.3 Single flexible link compound pendulum 52

3.3.4 Single flexible link with a rigid joint mass 64

3.3.5 General move of a three link arm 68

Chapter 4

Symbolic Generation of the Flexible Link Dynamics 81

Chapter 5

Summary and Conclusions 91

References 94

Appendix A

Denavit-Hartenberg Convention for Homogeneous Link Transformation Matrices 99

List of Tables

1.1 Computational burden of common dynamics formulations 8

3.1 Properties of the flexible link 45

3.2 Effect of the initial displacement on the period of oscillation of the rigid pendulum 50

4.1 Computational requirements for the inverse dynamics of a three link manipulator 87

4.2 Execution times for the inverse dynamics calculation. 89

List of Figures

2.1 Homogeneous transformation of coordinate frames 14

2.2 Homogeneous link tranformations 16

3.1 A flowchart of the FLX simulation 37

3.2 The first three modeshapes of a fixed free beam 41

3.3 The flexural modal coordinates of the FLX link 42

3.4 SRPEN geometry used to simulate the single rigid link pendulum . 46

3.5 FLX geometry used to simulate the single rigid link pendulum . . 48

3.6 Response of a single rigid link pendulum to an initial displacement 49

3.7 DRPEN geometry used to simulate the double link rigid pendulum 51

3.8 FLX geometry used to simulate the double link rigid pendulum . . 53

3.9 Response of a double rigid link pendulum to an initial displacement, simulated using DRPEN with a 1 ms timestep 54

3.10 Response of a double rigid link pendulum to an initial displacement, simulated using FLX with a 1 ms timestep 55

3.11 FPEN geometry used to model the single flexible link pendulum . 56

3.12 FLX geometry used to model the single flexible link pendulum . . 58

3.13 Response of a flexible link pendulum to an initial displacement and deflection, simulated using FPEN with a 10 ms timestep 59

3.14 The effect of varying integration time step on total system energy decay 60

3.15 Response of a flexible link pendulum to an initial displacement and deflection, simulated using FPEN with a 1 ms timestep 62

3.16 Response of a flexible link pendulum to an initial displacement and deflection, simulated using FLX with a 1 ms timestep 63

3.17 JMFLX geometry used to model a single flexible link with a rigid joint mass 65

3.18 FLX geometry used to model the single flexible link with a rigid joint mass 66

3.19 Response of a single flexible link with a rigid joint mass to an initial deflection of $\zeta = 0.012$, simulated using JMFLX with a 1 ms timestep 67

3.20 The effect of varying joint inertia on the frequency of link flexural oscillation 69

3.21 Response of the flexible link with a rigid joint mass to an initial deflection in the plane of the joint axis, simulated using FLX with a time step of 1 ms. 70

3.22 Three link manipulator 71

3.23 Command input to the three link manipulator 73

3.24 Initial and final positions of the three link manipulator 74

3.25 Response of the rigid link manipulator 75

3.26 Response of the flexible link manipulator: joint positions and system energy 76

3.27 Response of the flexible link manipulator: generalized deflection variables 77

3.28 Time history of the control joint torques 78

3.29 Time history of the joint torques from the inverse dynamics 79

A.1 The Denavit-Hartenberg convention for revolute joints 101

Chapter 1

Introduction

Dynamic simulation plays an important role in the design and operation of industrial robots. Dynamic simulation is useful in the design of manipulators, allowing components to be sized to accomplish a specific task or the relative merits of competing control systems to be studied [1]. Dynamic simulation is also important in offline programming, determining the feasibility of a move for a given manipulator and load combination to be determined without removing a robot from service. The tracking error of a manipulator following a trajectory can also be predicted, taking into account dynamic loading, with the use of simulation.

The dynamic equations of motion, central to any dynamic simulation, are also important in the control of robot motion. Given a joint trajectory, that is a time history of desired positions, velocities, and accelerations, the equations of motion in inverse dynamic form can be used to calculate the joint torques required to follow that trajectory. If these joint torques are applied to the manipulator, accurate trajectory following is possible. Luh and co-workers [2,3] have demonstrated the effectiveness of this technique in producing accurate trajectory following with

their "resolved acceleration" controller. However, this control method is very computationally intensive. These computations must be carried out repeatedly and quickly. Paul [4] states that the dynamic equations must be carried out at the servo rate for stable trajectory following, typically at least sixty times a second.

The quest for improved performance of robotic devices has prompted the desire to use lighter links. Lightweight manipulators promise the possibility of increased speed of operation, reduced material requirement, longer reach, reduced power consumption, and lowered mounting strength and rigidity requirement [5]. However lightweight manipulators have the disadvantages of excitation of flexural and torsional modes of oscillation by the control torques and a loss of static positioning [6] These effects increase both end effector position and tracking errors. Cycle time may also be adversely affected by the excitation of link oscillations. A move can be decomposed into two phases based on the type of control employed: the gross motion, where the load is brought as quickly as possible close to its final position typically using PD (proportional-derivative) control, and the final positioning of the load using PID (proportional-integral-derivative) and possibly compliant control. If there is vibration present in a manipulator after the gross motion of a move is complete, this residual vibration may interfere with the final positioning of the load. If this is the case, then the residual vibration must decay to within acceptable limits before final positioning of a load can be accomplished. The time required for residual vibration to decay may nullify any reduction in cycle time obtained through the use of lightweight flexible links. If the advantages of flexible link manipulators are to be realized in practice, the link oscillations must be controlled, placing increased demands on the control system.

There is an extensive literature on the modelling and control of flexible link manipulators [6,7,8,9,10,11]. However, much of the early work was in the control

of large flexible space structures, where the gross motions of the structure are relatively small and slow [12], and is not directly applicable to robotics, where the gross motions of the structure are large and fast. Although gains have been made in the control of linearized flexible robotic systems [5,7,9], the control of nonlinear flexible link manipulators remains an active area of research.

For the remainder of this chapter, the kinematics and dynamics of rigid link manipulators are presented as a starting point for the development of flexible link dynamics.

1.1 Rigid Link Manipulator Kinematics

Historically, robot kinematics has evolved from the kinematics of spatial linkages. Any robot can be considered to be a series of links connected by actuated joints. With a coordinate frame embedded in each link, the position and orientation of each link relative to its predecessor in the chain can be described using a homogeneous transformation. Pieper [13] was the first to apply homogeneous transformations to describe robot kinematics, using the Denavit-Hartenberg convention for homogeneous link transformations. The Denavit-Hartenberg convention is still the most popular means of describing robot kinematics and is presented in detail in Appendix A. The position and orientation of a link with respect to the base frame is just the product of the homogeneous transformations from the base to that link.

For each link, there is only a single generalized coordinate that describes the link position. If the joint is revolute, the generalized coordinate is the joint angle. For a prismatic joint, the generalized coordinate is the link extension. The generalized coordinate describing the motion of a joint is incorporated in the ho-

mogeneous transformation of the link, so that joint velocities and accelerations can be exressed as time derivatives of the homogeneous transformations.

The position of a point fixed in link i with respect to the coordinate frame of link j is ${}^{j}\boldsymbol{p}_i$. The prefixed superscript j implies that the position vector is expressed in the coordinates of frame j. If the defining coordinate frame was the base frame, that is $j = 0$, the prefixed superscript j is normally omitted. The postfixed subscript i implies that the point, whose position vector is ${}^{j}\boldsymbol{p}_i$, is fixed in link i. Using a homogeneous transformation of coordinates, the position of a point fixed in link i with respect to the base frame can be expressed as

$$\begin{aligned} {}^{0}\boldsymbol{p}_i &= {}^{0}\boldsymbol{W}_i\, {}^{i}\boldsymbol{p}_i \\ \boldsymbol{p}_i &= \boldsymbol{W}_i\, {}^{i}\boldsymbol{p}_i \end{aligned} \tag{1.1}$$

where

$\boldsymbol{W}_i$ is the cumulative transformation from the base frame to frame i and

${}^{i}\boldsymbol{p}_i$ is the position of the point fixed in link i expressed in link i's coordinates.

Since the point is fixed in link i, the position vector ${}^{i}\boldsymbol{p}_i$ is a constant vector and the velocity of that point in the base frame is

$$\begin{aligned} \dot{\boldsymbol{p}}_i &= \dot{\boldsymbol{W}}_i\, {}^{i}\boldsymbol{p}_i \\ &= \left\{ \sum_{j=1}^{i} \frac{\partial \boldsymbol{W}_i}{\partial q_j} \dot{q}_j \right\} {}^{i}\boldsymbol{p}_i \end{aligned} \tag{1.2}$$

where q_j is the generalized coordinate describing the motion of link j.

1.2 Rigid Link Manipulator Dynamics

Manipulator dynamic equations have been developed chiefly using one of two approaches. Either a Lagrangian formulation or a Newton-Euler formulation,

based on D'Alembert's principle, has been employed. The Lagrangian formulation based on 4×4 transformation matrices will be presented in some detail, as it relates to the development of flexible link dynamics.

Uicker [14] first developed the dynamics of a spatial linkage using Lagrangian mechanics and 4 × 4 transformation matrices to express the kinematics. This formulation was later particularized by Kahn [15] for serial open loop chains. The form of the inverse dynamics solution is

$$F_i = \sum_{j=i}^{n} \left[\sum_{k=1}^{j} \left\{ Tr \left(\frac{\partial W_j}{\partial q_i} J_j \frac{\partial W_j^T}{\partial q_k} \right) \ddot{q}_k \right\} + \sum_{k=1}^{j} \sum_{l=1}^{j} \left\{ Tr \left(\frac{\partial W_j}{\partial q_i} J_j \frac{\partial^2 W_j^T}{\partial q_k \partial q_l} \right) \dot{q}_k \dot{q}_l \right\} - m_j g^T \frac{\partial W_j}{\partial q_i} r_j \right] \tag{1.3}$$

where

F_i is the generalized force or torque,

W_j is the cumulative transformation from the base frame to frame j,

q_j is the generalized coordinate for link j,

J_j is the inertia tensor for link j with respect to the proximal joint of link j in link j's body fixed coordinates,

m_j is the mass of link j,

g is the gravity vector,

r_j is the position of the centre of mass of link j in frame j,

n is the number of links in the chain,

Tr is the trace operator, and

W_j^T is the transpose of W_j.

The complexity of this equation arises from the dynamic coupling between the links. The joint force or torque at a joint depends on the state of all the joints as indicated by the summations in equation (1.3).

Waters [16] expressed the generalized velocities and accelerations, $\dot{W}_j$ and $\ddot{W}_j$, in a recursive form. Hollerbach [17] further simplified the recursion relations such that the inverse dynamics solution could be written as

$$F_i = \sum_{j=i}^{n} \left[Tr \left\{ \frac{\partial W_j}{\partial q_i} J_j \ddot{W}_j^T \right\} - m_j g^T \frac{\partial W_j}{\partial q_i} r_j \right]. \tag{1.4}$$

The specific recursion relations used by Hollerbach [17] are

$$\begin{aligned} W_j &= W_{j-1} A_j \\ \dot{W}_j &= \dot{W}_{j-1} A_j + W_{j-1} \frac{\partial A_j}{\partial q_j} \dot{q}_j \\ \ddot{W}_j &= \ddot{W}_{j-1} A_j + 2\dot{W}_{j-1} \frac{\partial A_j}{\partial q_j} \dot{q}_j + W_{j-1} \left(\frac{\partial^2 A_j}{\partial q_j^2} \dot{q}_j^2 + \frac{\partial A_j}{\partial q_j} \ddot{q}_j \right) \end{aligned} \tag{1.5}$$

where the starting values for the recursion are $W_0 = \{1\}$ and $\dot{W}_0 = \ddot{W}_0 = \{0\}$ and A_j is the link transformation for link j. Hollerbach further noted that the partial derivative with respect to the joint variable q_j of the transformation W_j could be written as

$$\frac{\partial W_j}{\partial q_i} = \frac{\partial W_i}{\partial q_i} {}^{i}W_j. \tag{1.6}$$

Using equation (1.6), the generalized force equation could be written as

$$\begin{aligned} F_i &= \sum_{j=i}^{n} \left[Tr \left\{ \frac{\partial W_i}{\partial q_i} {}^{i}W_j J_j \ddot{W}_j^T \right\} - m_j g^T \frac{\partial W_i}{\partial q_i} {}^{i}W_j r_j \right] \\ &= Tr \left\{ \frac{\partial W_i}{\partial q_i} \sum_{j=i}^{n} {}^{i}W_j J_j \ddot{W}_j^T \right\} - g^T \frac{\partial W_i}{\partial q_i} \sum_{j=i}^{n} m_j {}^{i}W_j r_j. \end{aligned} \tag{1.7}$$

And the following recursions could be used

$$\begin{aligned} \boldsymbol{D}_i &= \sum_{j=i}^{n} {}^i\boldsymbol{W}_j \boldsymbol{J}_j \tilde{\boldsymbol{W}}_j^T \\ &= {}^i\boldsymbol{W}_i \boldsymbol{J}_i \tilde{\boldsymbol{W}}_i^T + \sum_{j=i+1}^{n} \boldsymbol{A}_{i+1}\, {}^{i+1}\boldsymbol{W}_j \boldsymbol{J}_j \tilde{\boldsymbol{W}}_j^T \\ &= \boldsymbol{J}_i \tilde{\boldsymbol{W}}_i^T + \boldsymbol{A}_{i+1} \boldsymbol{D}_{i+1} \end{aligned} \tag{1.8}$$

and

$$\begin{aligned} \boldsymbol{c}_i &= \sum_{j=i}^{n} m_j\, {}^i\boldsymbol{W}_j \boldsymbol{r}_j \\ &= m_i \boldsymbol{r}_i + \boldsymbol{A}_{i+1} \boldsymbol{c}_{i+1} \end{aligned} \tag{1.9}$$

where $\boldsymbol{D}_{n+1} = \boldsymbol{c}_{n+1} = \{0\}$. Using these recursions the generalized force becomes

$$F_i = Tr\left\{ \frac{\partial \boldsymbol{W}_i}{\partial q_i} \boldsymbol{D}_i \right\} - \boldsymbol{g}^T \frac{\partial \boldsymbol{W}_i}{\partial q_i} \boldsymbol{c}_i. \tag{1.10}$$

The $\tilde{\boldsymbol{W}}_i^T$ terms are computed first using equation (1.5), starting from the base of the manipulator and working towards the end effector. Then the $\boldsymbol{D}_i$ and $\boldsymbol{c}_i$ are computed using equations (1.8) and (1.9) starting from the end effector and working towards the base of the manipulator.

The computational burden of the common formulations of the rigid body inverse dynamics equations for a six degree of freedom arm are compared in Table 1.1. The computational burden of the inverse dynamics solution has been reduced over the original Uicker-Kahn formulation due to the recursive nature of Hollerbach's formulation. The original Uicker-Kahn formulation of the rigid link dynamics is the most computationally intensive. The number of operations, that is multiplications and additions, in the inverse dynamics calculation using this formulation is of order n^4, where n is the number of links. In contrast, the

Number of operations × = multiplication, + = addition			
Method	Operation	Degrees of Freedom n	6
Uicker-Kahn	×	$25n^4 + 66n^3 + 129n^2 + 42n - 96$	67984
	+	$32n^4 + 86n^3 + 171n^2 + 53n - 128$	51456
Hollerbach	×	$830n - 592$	4388
(4×4)	+	$675n - 464$	3586
Hollerbach	×	$412n - 277$	2195
(3×3)	+	$320n - 201$	1719
Newton-Euler	×	$150n - 48$	852
	+	$131n - 48$	738

Table 1.1: Computational burden of common dynamics formulations (data from Brady [22])

number of operations in the inverse dynamics calculation using either of Hollerbach's recursive formulations or the Newton-Euler formulation is of order n. The use of Hollerbach's recursive formulation of the rigid link dynamics has reduced the comutational burden of the inverse dynamics calculation by $\sim 93\%$ over the original Uicker-Kahn formulation. The Newton-Euler formulation of the rigid link dynamics has been employed by Hooker and Margulies [18], Stepanenko and Vukobratovic [19], Orin and coworkers [20], and Luh, Walker, and Paul [21]. The Newton-Euler formulation is inherently more efficient then the Lagrangian formulations, as shown in Table 1.1, involving 20% of the number of operations required in Hollerbach's 4×4 matrix Lagrangian formulation. This increase in computational efficiency is due to the use of 3×3 rotation transformations and vector translations rather than the 4×4 homogeneous transformations used in the Lagrangian formulations. If 3×3 rotation transformations are used in the Lagrangian formulation, the computational efficiency is still inferior to the Newton-Euler formulation. The 3×3 matrix Lagrangian formulation requires 2.6 times the number of computations required by the Newton-Euler formulation.

1.3 Symbolic Generation of Dynamics Equations

Recently, impressive gains in the computational efficiency of the rigid link dynamics have been reported [23] through the use of symbolic generation of the equations of motion applied to the Lagrangian formulation of the dynamics. Toogood [23] has reported the fewest number of arithmetic operations to date for the evaluation of the symbolically generated dynamics equations of the Stanford manipulator.

In symbolic generation of the dynamics equations, the matrix multiplications

involved in the Lagrangian formulation using the 4×4 homogeneous transformations are performed only once using symbols. While convenient, the 4×4 transformations in the Lagrangian formulation are inefficient because of the presence of zeroes and ones. Using symbolic manipulation, needless operations such as multiplication by 1, -1, or 0 and addition of 0 are not performed, instead they are replaced by appropriate assignments. The use of the trace operator in the Lagrangian formulation results in the evaluation of matrix elements which are not required for evaluation of the dynamics. Through the use of symbolic generation, the evaluation of these unused matrix elements can be avoided.

The increase in computational efficiency through the use of symbolic generation of the dynamics equations is due to more than just a reduction in the number of floating point operations required. The symbolically generated dynamics are encoded in a subroutine as a sequence of scalar operations. This eliminates the overhead of repeated calls to matrix multiplier, trace, and transpose subroutines.

1.4 Outline of This Work

The equations of motion of flexible link manipulators have been formulated using Lagrange's equations [6,7,8,9,10,11] and Kane's dynamics equations [24,25]. The elastic deformation in the flexible link dynamics has been modelled by finite element techniques [6,12,26] and by the method of assumed modes [7,8,25]. The inclusion of link flexibility into the model of a manipulator, greatly increases the complexity of the dynamics equations. However, the computational burden of the flexible link dynamics has not been reported in the literature as it has been for the rigid link dynamics. Also, symbolic generation of the flexible link dynamics has not been reported.

In Chapter 2 of this thesis, the formulation of the flexible link dynamics following the development due to Book [27] is presented. Lagrange's equations, using the 4×4 homogeneous transformations to describe the kinematics, are used to formulate the equations of motion. The link flexibility is incorporated into the model using the method of assumed modes.

A numerical model, called FLX, based on the flexible link dynamics presented in Chapter 2 has been written. In Chapter 3, an overview of this simulator and some attempts at verifying its' correctness are presented.

In Chapter 4, the symbolic generation of the flexible link dynamics, using the techniques of Toogood [23] and Kermack [28], is presented. The computational burden of the numerical model, FLX, is compared to that of the symbolically generated dynamics equations.

Finally in Chapter 5, the results of this investigation are summarized.

Chapter 2

Derivation of the Equations of Motion

In this chapter the kinematics and dynamics for a serial chain of flexible links are described. The link kinematics will first be described and then the formulation of the Lagrangian dynamics will be outlined. This formulation is similar to the Uiker-Kahn [14,29] formulation of the dynamics of a chain of rigid links modified to incorporate link flexibility, following the method of Book [27].

In this analysis, the flexure and torsion of the links is described using the method of assumed modes. Small deflections and rotations, that can be described by a summation of modal shapes and linear elasticity, are assumed. The usual strength of materials assumptions are assumed in the calculation of the strain energy of bending and torsion [30,31]. The effects of rotary inertia and shear deformation in bending are ignored, so this analysis is only applicable to the first few modes of vibration of long, thin links.

2.1 Kinematics

At this point, the kinematics of a serial link manipulator will be described. A serial link manipulator is a sequence of links connected by actuated joints. The base link of the chain is a fixed inertial reference frame. Each link in the chain has a joint at its proximal end. In this analysis only revolute joints are considered; the only permissible motion of a revolute joint is rotation about its joint axis. In order to describe the kinematics of an n flexible link serial chain, $\sum_{i=1}^{n}(m_i + 1)$ generalized coordinates are required: one for each link's joint motion and one for each of the m_i modes of link deflection.

The kinematics of both the revolute joint motion and link deformation are described using modified Denavit - Hartenberg 4 x 4 transform matrices. The reader who is unfamiliar with homogeneous coordinate transformations, and in particular with the Denavit - Hartenberg convention [32] for the representation of a rigid link by an homogeneous transformation matrix, should refer to Appendix A. By a homogeneous transformation of coordinates, the position of a point in frame i can be described in any other coordinate system j if the transformation ${}^{j}W_i$ is known. The form of this transformation matrix is

$$ {}^{j}W_i = \begin{pmatrix} & & & x_j \text{ component of } O_i \\ & {}^{j}R_i & & y_j \text{ component of } O_i \\ & & & z_j \text{ component of } O_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.1} $$

where ${}^{j}R_i$ is a 3 x 3 matrix of direction cosines describing the orientation of frame i with respect to frame j and the fourth column of the transformation contains the position of the origin O_i of frame i in frame j.

If ${}^{i}h_i$ is the position vector of a point in frame i, shown in Figure 2.1, then

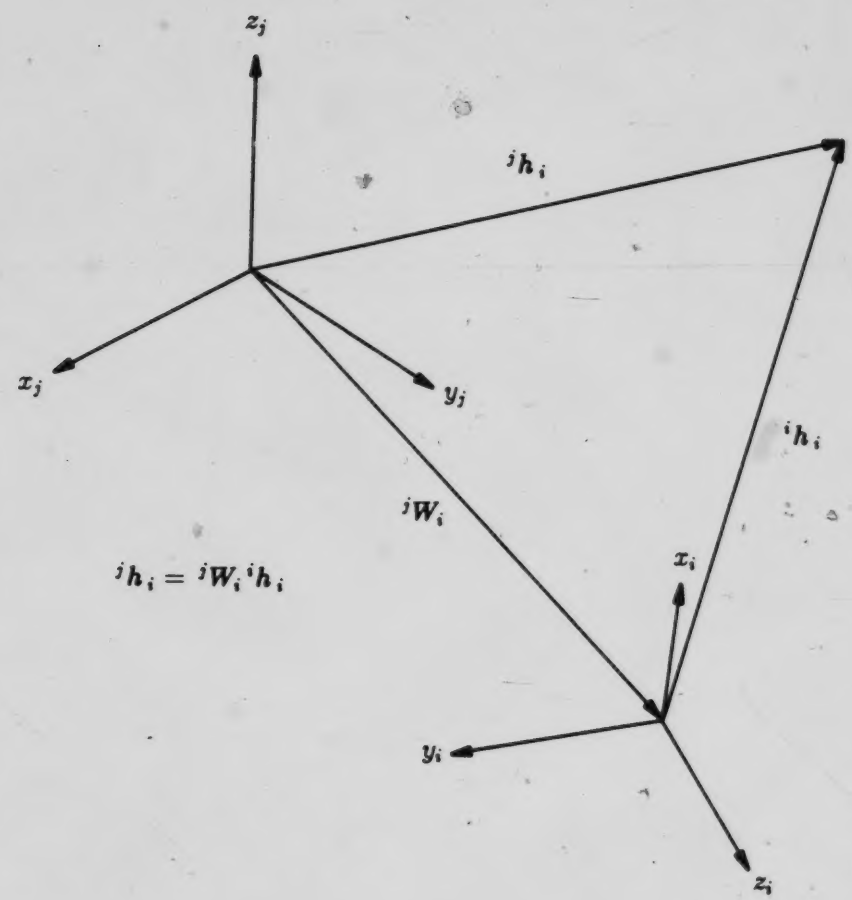

Figure 2.1: Homogeneous transformation of coordinate frames

the position vector of that point in frame j, the previous frame, is

$$ {}^{j}\boldsymbol{h}_i = {}^{j}\boldsymbol{W}_i \, {}^{i}\boldsymbol{h}_i \tag{2.2} $$

where ${}^{j}\boldsymbol{W}_i$ is the homogeneous transformation from frame j to frame i. With a serial chain of links, the position vector of a point on link i, in terms of the fixed inertial coordinate system of the base, is given by the product of the individual link transformations from the base to link i multiplied by the position vector of that point in frame i,

$$ \begin{aligned} \boldsymbol{h}_i &= {}^{0}\boldsymbol{h}_i \\ &= {}^{0}\boldsymbol{W}_1 \, {}^{1}\boldsymbol{W}_2 \ldots {}^{i-1}\boldsymbol{W}_i \, {}^{i}\boldsymbol{h}_i \\ &= {}^{0}\boldsymbol{W}_i \, {}^{i}\boldsymbol{h}_i \\ &= \boldsymbol{W}_i \, {}^{i}\boldsymbol{h}_i. \end{aligned} \tag{2.3} $$

The prefixed superscript 0 of ${}^{0}\boldsymbol{h}_i$ and ${}^{0}\boldsymbol{W}_i$ refers to the base frame and is normally omitted.

In order to incorporate the distributed link flexibility in the kinematics, the homogeneous link transformation is decomposed into three parts shown in Figure 2.2. The $\boldsymbol{A}$ matrix describes the rotation of the joint. This is not the standard Denavit-Hartenberg $\boldsymbol{A}$ matrix described in Appendix A, it is just the rotation transformation for the joint. A constant $\boldsymbol{S}$ matrix allows for a rigid joint mass, and an $\boldsymbol{L}$ matrix describes the bend and twist at the distal end relative to the proximal end of the flexible link due to the distributed flexibility. Thus, the cumulative transformation from the base up to and including joint j can be written as

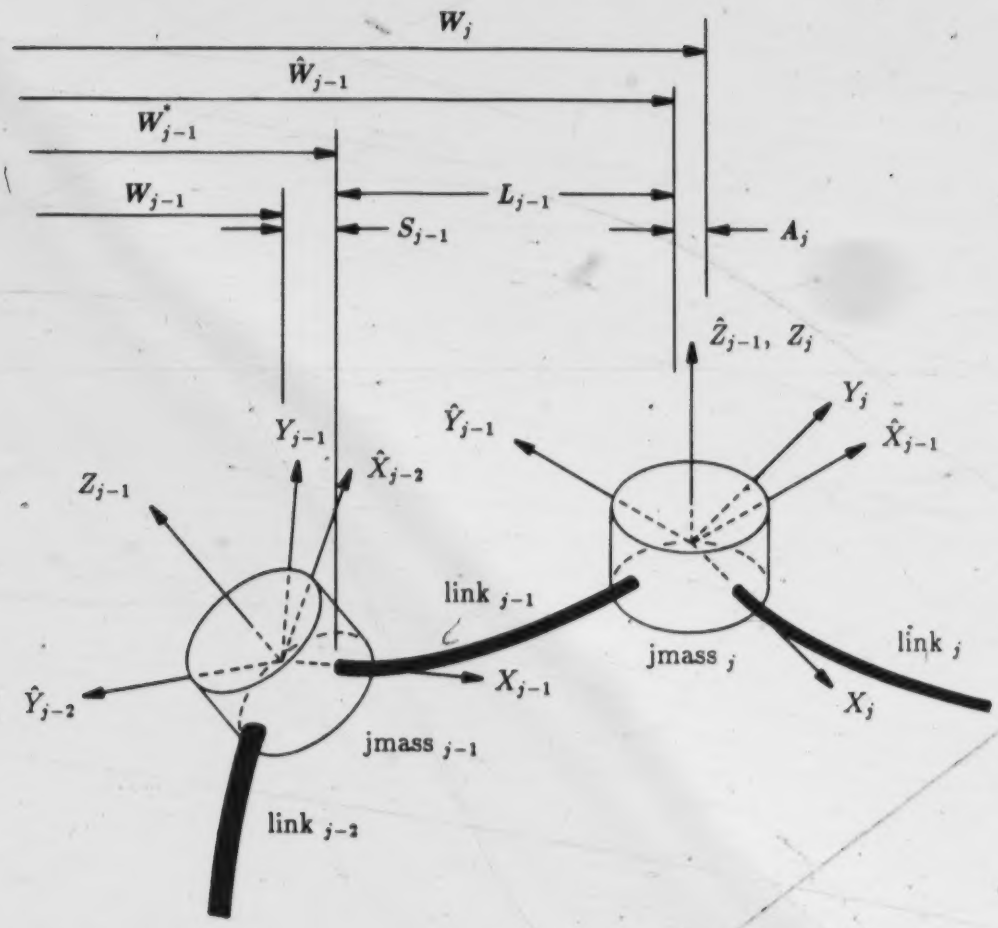

Figure 2.2: Homogeneous link tranformations

$$\begin{aligned} \boldsymbol{W}_j &= \boldsymbol{W}_{j-1}\boldsymbol{S}_{j-1}\boldsymbol{L}_{j-1}\boldsymbol{A}_j \\ &= \boldsymbol{W}^{*}_{j-1}\boldsymbol{L}_{j-1}\boldsymbol{A}_j \\ &= \hat{\boldsymbol{W}}_{j-1}\boldsymbol{A}_j \end{aligned} \tag{2.4}$$

where

$\boldsymbol{W}_{j-1}$ is the cumulative transformation from the base to joint $j-1$,

$\boldsymbol{S}_{j-1}$ is the rigid joint mass transformation from the joint axis to the base of the flexible link,

$\boldsymbol{L}_{j-1}$ is the link transformation from the base of the link $j-1$ to the joint j,

$\boldsymbol{A}_j$ is the rotation transformation for joint j,

$\boldsymbol{W}^{*}_{j-1}$ is the cumulative transformation from the base to the proximal end of the flexible link $j-1$, and

$\hat{\boldsymbol{W}}_{j-1}$ is the cumulative transformation from the base to the distal end of link $j-1$.

If there is no deformation in link $j-1$, then $\hat{X}_{j-1}$ coincides with X_{j-1} and $\hat{Y}_{j-1}$ and $\hat{Z}_{j-1}$ are parallel to Y_{j-1} and Z_{j-1} respectively.

To incorporate the deflection of a link, the approach of modal analysis [33] is used, in which the deformed shape of a link can be described as a sum of normal modes. The position vector of a point on a flexible link i in frame i can be written as

$$^{i}\boldsymbol{h}_i = \begin{pmatrix} \eta \\ 0 \\ 0 \\ 1 \end{pmatrix} + \sum_{j=1}^{m_i} \delta_{ij} \begin{pmatrix} x_{ij}(\eta) \\ y_{ij}(\eta) \\ z_{ij}(\eta) \\ 1 \end{pmatrix} \tag{2.5}$$

where

$x_{ij}(\eta)$, $y_{ij}(\eta)$, and $z_{ij}(\eta)$ are X_i, Y_i, and Z_i displacement components respectively of mode j on link i evaluated at a distance η from the base of the link,

δ_{ij} is the time varying amplitude or modal coordinate of mode j on link i, and

m_i is the number of modes used to describe the deflection of link i.

The link transformation matrix $\boldsymbol{L}$ must take into account the rotations at the end of a link as well as the displacements as a summation of modes. If the restriction of small rotations is applied, the rotations can be assumed to add vectorially and the direction cosine matrix simplifies [34]. The link transformation matrix can then be written as

$$\boldsymbol{L}_i = \boldsymbol{H}_i + \sum_{j=1}^{m_i} \delta_{ij} \boldsymbol{M}_{ij} \tag{2.6}$$

where

$$\boldsymbol{H}_i = \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.7}$$

and

$$\boldsymbol{M}_{ij} = \begin{pmatrix} 0 & -\theta_{zij}(a_i) & \theta_{yij}(a_i) & x_{ij}(a_i) \\ \theta_{zij}(a_i) & 0 & -\theta_{xij}(a_i) & y_{ij}(a_i) \\ -\theta_{yij}(a_i) & \theta_{xij}(a_i) & 0 & z_{ij}(a_i) \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.8}$$

and where the $\theta_{xij}(a_i)$, $\theta_{yij}(a_i)$ and, $\theta_{zij}(a_i)$ are the X_i, Y_i and, Z_i rotation components of mode j on link i respectively, evaluated at a_i where a_i is the length of link i.

Since the time derivatives of the generalized coordinates are required for the formulation of Lagrange's equations, expressions for the time derivatives of the transformation matrices will now be developed.

The position and velocity of a point on the flexible link i in the inertial base frame are

$$\begin{aligned} \boldsymbol{h}_i &= \boldsymbol{W}_i \boldsymbol{S}_i {}^i\boldsymbol{h}_i \\ &= \boldsymbol{W}_i^* {}^i\boldsymbol{h}_i \end{aligned} \tag{2.9}$$

and

$$\begin{aligned} \dot{\boldsymbol{h}}_i &= \dot{\boldsymbol{W}}_i \boldsymbol{S}_i {}^i\boldsymbol{h}_i + \boldsymbol{W}_i \boldsymbol{S}_i {}^i\dot{\boldsymbol{h}}_i \\ &= \dot{\boldsymbol{W}}_i^* {}^i\boldsymbol{h}_i + \boldsymbol{W}_i^* {}^i\dot{\boldsymbol{h}}_i \end{aligned} \tag{2.10}$$

respectively. The position and velocity of a point on the rigid proximal mass of link i in the inertial base frame are

$$\boldsymbol{p}_i = \boldsymbol{W}_i {}^i\boldsymbol{p}_i \tag{2.11}$$

and

$$\dot{\boldsymbol{p}}_i = \dot{\boldsymbol{W}}_i {}^i\boldsymbol{p}_i \tag{2.12}$$

respectively. The position vector ${}^i\boldsymbol{p}_i$ is a constant since the joint mass is assumed to be rigid. With a serial chain of links, it is computationally efficient to relate the position and velocity of a point on a link to preceding members in the chain, in a manner similar to Hollerbach's [17] method for rigid links. Hollerbach [17] claims that, for rigid link manipulators, the application of Waters [16] backward

recursion of the accelerations $\bar{W}_j$ results in a reduction of the number of floating point operations from order n^4 to n^2 dependence, where n is the number of links. Differentiating equation (2.4), gives

$$\dot{W}_j = \dot{\hat{W}}_{j-1} A_j + \hat{W}_{j-1} \frac{\partial A_j}{\partial q_j} \dot{q}_j \tag{2.13}$$

and

$$\bar{W}_j = \bar{\hat{W}}_{j-1} A_j + 2\dot{\hat{W}}_{j-1} \frac{\partial A_j}{\partial q_j} \dot{q}_j + \hat{W}_{j-1} \left(\frac{\partial^2 A_j}{\partial q_j^2} \dot{q}_j^2 + \frac{\partial A_j}{\partial q_j} \bar{q}_j \right) \tag{2.14}$$

where the starting values for the recursion are $W_0 = \dot{W}_0 = \bar{W}_0 = \{0\}$ and q_j is the joint variable of joint j. The transformation $\hat{W}_j$ and its derivatives can also be computed recursively as follows:

$$\hat{W}_j = W_j^* L_j \tag{2.15}$$

differentiating equation (2.15), using equation (2.6), gives

$$\dot{\hat{W}}_j = \dot{W}_j^* L_j + W_j^* \sum_{k=1}^{m_j} \dot{\delta}_{jk} M_{jk} \tag{2.16}$$

and

$$\bar{\hat{W}}_j = \bar{W}_j^* L_j + 2\dot{W}_j^* \sum_{k=1}^{m_j} \dot{\delta}_{jk} M_{jk} + W_j^* \sum_{k=1}^{m_j} \bar{\delta}_{jk} M_{jk} \tag{2.17}$$

where $\hat{W}_0 = \{1\}$, $\dot{\hat{W}}_0 = \bar{\hat{W}}_0 = \{0\}$,

$$\dot{W}_j^* = \dot{W}_j S_j, \tag{2.18}$$

and

$$\bar{W}_j^* = \bar{W}_j S_j \tag{2.19}$$

since S_j is a constant transformation. The separation of the joint transformation A_j from the link transformation L_j has increased the complexity of the recursive scheme for the acceleration transformations over the rigid link case, resulting in coupled recursion for $\bar{W}_j$ and $\bar{\hat{W}}_j$.

2.2 Dynamics

Having made these modifications to the rigid link kinematics, it is a simple matter to formulate the kinetic and potential energy for each link and then using Lagrange's equations to formulate the equations of motion for the manipulator in terms of the generalized coordinates.

2.2.1 Kinetic Energy

In this section, an expression for the kinetic enegy of the system is developed. The total system kinetic energy is composed of two parts

$$T = T^f + T^r$$

where

T^f is the contribution to the system kinetic energy arising from the flexible portion of a link and

T^r is the contribution to the system kinetic energy arising from the rigid joint mass.

First, the kinetic energy of a differential element of a link is formulated. This is integrated over the length of each link and summed over all the links to give the total system kinetic energy.

The kinetic energy of a point on the flexible link i is

$$dT_i^f = \frac{1}{2}\,\mu(\eta)d\eta\; Tr\left\{\,\dot{\boldsymbol{h}}_i\,\dot{\boldsymbol{h}}_i^T\,\right\}. \tag{2.20}$$

Using equation (2.10) the kinetic energy of a point on the flexible link i can be written as

$$dT_i^f = \frac{1}{2}\,\mu(\eta)d\eta\; Tr\left\{ \dot{W}_i^{*}\,{}^{i}\dot{h}_i\,{}^{i}\dot{h}_i^{T}\;\dot{W}_i^{*T} + 2\,\dot{W}_i^{*}\,{}^{i}\dot{h}_i\,{}^{i}\dot{h}_i^{T}\,W_i^{*T} + W_i^{*}\,{}^{i}\dot{h}_i\,{}^{i}\dot{h}_i^{T}\,W_i^{*T} \right\} \tag{2.21}$$

where

$${}^{i}\dot{h}_i = \sum_{j=1}^{m_i} \dot{\delta}_{ij} \begin{pmatrix} x_{ij}(\eta) \\ y_{ij}(\eta) \\ z_{ij}(\eta) \\ 0 \end{pmatrix}, \tag{2.22}$$

$\mu(\eta)$ is the mass per unit length of the link, and

$d\eta$ is a differential length of the link.

The kinetic energy of a point on the rigid proximal mass of link i is

$$\begin{aligned} dT_i^r &= \frac{1}{2}\,dm\; Tr\left\{ \dot{p}_i\;\dot{p}_i{}^{T} \right\} \\ &= \frac{1}{2}\,dm\; Tr\left\{ \dot{W}_i\,{}^{i}p_i\,{}^{i}p_i^{T}\,\dot{W}_i^{T} \right\}. \end{aligned} \tag{2.23}$$

Integrated over the length of the link and summed over the number of links, the total kinetic energy of the arm is

$$T = \sum_{i=1}^{n} \left[Tr\left\{ \dot{W}_i^{*} B_{3i}\,\dot{W}_i^{*T} + 2\,\dot{W}_i^{*} B_{2i} W_i^{*T} + W_i^{*} B_{1i} W_i^{*T} + \dot{W}_i J_{Ri} \dot{W}_i^{T} \right\} \right] \tag{2.24}$$

where

$$\begin{aligned} B_{1i} &= \frac{1}{2}\int_0^{a_i} \mu(\eta)\,{}^{i}\dot{h}_i\,{}^{i}\dot{h}_i^{T}\,d\eta \\ &= \sum_{j=1}^{m_i}\sum_{k=1}^{m_i} \dot{\delta}_{ik}\dot{\delta}_{ij} C_{ikj} \end{aligned} \tag{2.25}$$

$$B_{2i} = \frac{1}{2}\int_0^{a_i} \mu(\eta)\, {}^i\boldsymbol{h}_i\, {}^i\dot{\boldsymbol{h}}_i^T\, d\eta$$

$$= \sum_{j=1}^{m_i} \dot{\delta}_{ij} C_{ij} + \sum_{k=1}^{m_i}\sum_{j=1}^{m_i} \delta_{ik}\dot{\delta}_{ij} C_{ikj} \tag{2.26}$$

$$B_{3i} = \frac{1}{2}\int_0^{a_i} \mu(\eta)\, {}^i\boldsymbol{h}_i\, {}^i\boldsymbol{h}_i^T\, d\eta$$

$$= C_i + \sum_{j=1}^{m_i} \delta_{ij}\left(C_{ij} + C_{ij}^T\right) + \sum_{k=1}^{m_i}\sum_{j=1}^{m_i} \delta_{ik}\delta_{ij} C_{ikj} \tag{2.27}$$

$$J_{Ri} = \frac{1}{2}\int_{\text{link}_i} {}^i\boldsymbol{p}_i\, {}^i\boldsymbol{p}_i^T\, dm \tag{2.28}$$

and

$$C_i = \frac{1}{2}\int_0^{a_i} \mu(\eta)(\eta\ 0\ 0\ 1)^T(\eta\ 0\ 0\ 1)d\eta \tag{2.29}$$

$$C_{ij} = \frac{1}{2}\int_0^{a_i} \mu(\eta)(\eta\ 0\ 0\ 1)^T(x_{ij}\ y_{ij}\ z_{ij}\ 0)d\eta \tag{2.30}$$

$$C_{ikj} = \frac{1}{2}\int_0^{a_i} \mu(\eta)(x_{ik}\ y_{ik}\ z_{ik}\ 0)^T(x_{ij}\ y_{ij}\ z_{ij}\ 0)d\eta. \tag{2.31}$$

After performing the required differentiations, the kinetic energy terms in Lagrange's equations for the joint position variables may be written as

$$\frac{d}{dt}\left(\frac{\partial T^I}{\partial \dot{q}_j}\right) - \frac{\partial T^I}{\partial q_j} =$$

$$2\sum_{i=j}^{n} Tr\left[\frac{\partial \boldsymbol{W}_i^*}{\partial q_j}\left\{G_i \ddot{\boldsymbol{W}}_i^{*T} + 2\sum_{k=1}^{m_i}\dot{\delta}_{ik} D_{ik}\dot{\boldsymbol{W}}_i^{*T} + \sum_{k=1}^{m_i}\ddot{\delta}_{ik} D_{ik}\boldsymbol{W}_i^{*T}\right\}\right] \tag{2.32}$$

and

$$\frac{d}{dt}\left(\frac{\partial T^r}{\partial \dot{q}_j}\right) - \frac{\partial T^r}{\partial q_j} = 2\sum_{i=j}^{n} Tr\left[\frac{\partial \boldsymbol{W}_i}{\partial q_j} J_{Ri}\ddot{\boldsymbol{W}}_i^T\right]. \tag{2.33}$$

The kinetic energy term in Lagrange's equations for the rigid joint mass, equation (2.33), corresponds to the first term in equation (1.4) of Hollerbach's formulation of the rigid link dynamics. The two terms appear to differ by a factor of 2,

but this is due to a difference in the definition of the inertia tensor. The inertia tensor J_{Ri} is $\frac{1}{2}$ Hollerbach's inertia tensor.

The kinetic energy terms in Lagrange's equations for the modal deflection amplitudes may be written as

$$\frac{d}{dt}\left(\frac{\partial T^f}{\partial \dot{\delta}_{jf}}\right) - \frac{\partial T^f}{\partial \delta_{jf}} = \left(2\sum_{i=j+1}^{n} Tr\left[\frac{\partial W_i^*}{\partial \delta_{jf}}\left\{G_i \dot{W}_i^{*T} + 2\sum_{k=1}^{m_i}\dot{\delta}_{ik}D_{ik}\dot{W}_i^{*T} + \sum_{k=1}^{m_i}\bar{\delta}_{ik}D_{ik}W_i^{*T}\right\}\right]\right) + 2\,Tr\left[\left\{\bar{W}_j^* D_{jf} + 2\,\dot{W}_j^*\sum_{k=1}^{m_j}\dot{\delta}_{jk}C_{jkf} + W_j^*\sum_{k=1}^{m_j}\bar{\delta}_{jk}C_{jkf}\right\}W_j^{*T}\right] \tag{2.34}$$

and

$$\frac{d}{dt}\left(\frac{\partial T^r}{\partial \dot{\delta}_{jf}}\right) - \frac{\partial T^r}{\partial \delta_{jf}} = 2\sum_{i=j+1}^{n} Tr\left[\frac{\partial W_i}{\partial \delta_{jf}} J_{Ri}\bar{W}_i^T\right] \tag{2.35}$$

where

$$G_i = B_{3i} \tag{2.36}$$

and

$$D_{ik} = C_{ik} + \sum_{l=1}^{m_i}\delta_{il}C_{ikl}. \tag{2.37}$$

2.2.2 Potential Energy

The potential energies considered in this analysis are the gravitational potential and the strain energy of the flexible links in bending and torsion. Postfixed subscripts g and e are used to distinguish between gravitational and elastic potential energies respectively. Postfixed superscripts f and r are used to distinguish between the contributions to the total potential energy arising from the flexible portion of a link and the rigid proximal mass respectively.

The gravitational potential of a flexible link i can be expressed as

$$\begin{aligned} V_{gi}^{f} &= -g^T W_i^* \int_0^{a_i} \mu(\eta)\, {}^i h_i\, d\eta \\ &= -g^T W_i^* r_i \end{aligned} \tag{2.38}$$

where

$$r_i = \text{mass}_i \begin{pmatrix} \text{cg}_i \\ 0 \\ 0 \\ 1 \end{pmatrix} + \sum_{j=1}^{m_i} \delta_{ij} e_{ij}, \tag{2.39}$$

$$\text{cg}_i = \frac{1}{\text{mass}_i} \int_0^{a_i} \mu(\eta)\eta\, d\eta, \tag{2.40}$$

$$e_{ij} = \int_0^{a_i} \mu(\eta) \begin{pmatrix} x_{ij} \\ y_{ij} \\ z_{ij} \\ 0 \end{pmatrix} d\eta, \tag{2.41}$$

and

mass$_i$ is the mass of the flexible link i and

cg$_i$ is the distance from the base of the flexible link to its centre of mass when the link is undeformed.

The gravitational potential of the rigid proximal mass jmass$_i$ of link i is

$$V_{gi}^{r} = -g^T W_i M_i^r \tag{2.42}$$

where

$$M_i^{rT} = (0\ 0\ 0\ \text{jmass}_i). \tag{2.43}$$

The total gravitational potential of the manipulator is then

$$V_g = -g^T \sum_{i=1}^{n} \left\{ W_i^* r_i + W_i M_i^r \right\}. \tag{2.44}$$

The required derivatives of the gravitational potential are

$$\begin{aligned}\frac{\partial V_g^f}{\partial q_j} &= -\boldsymbol{g}^T \frac{\partial}{\partial q_j}\left(\sum_{i=1}^{n} \boldsymbol{W}_i^* \boldsymbol{r}_i\right) \\ &= -\boldsymbol{g}^T \sum_{i=j}^{n} \frac{\partial \boldsymbol{W}_i^*}{\partial q_j} \boldsymbol{r}_i \end{aligned} \tag{2.45}$$

$$\begin{aligned}\frac{\partial V_g^r}{\partial q_j} &= -\boldsymbol{g}^T \frac{\partial}{\partial q_j}\left(\sum_{i=1}^{n} \boldsymbol{W}_i \boldsymbol{M}_i^r\right) \\ &= -\boldsymbol{g}^T \sum_{i=j}^{n} \frac{\partial \boldsymbol{W}_i}{\partial q_j} \boldsymbol{M}_i^r \end{aligned} \tag{2.46}$$

$$\begin{aligned}\frac{\partial V_g^f}{\partial \delta_{jf}} &= -\boldsymbol{g}^T \frac{\partial}{\partial \delta_{jf}}\left(\sum_{i=1}^{n} \boldsymbol{W}_i^* \boldsymbol{r}_i\right) \\ &= \begin{cases} -\boldsymbol{g}^T \left[\displaystyle\sum_{i=j+1}^{n} \left(\frac{\partial \boldsymbol{W}_i^*}{\partial \delta_{jf}} \boldsymbol{r}_i\right) - \boldsymbol{W}_j^* \boldsymbol{e}_{jf}\right], & \text{for } 1 \le j < n; \\ -\boldsymbol{g}^T \boldsymbol{W}_n^* \boldsymbol{e}_{nf}, & \text{for } j = n. \end{cases} \end{aligned} \tag{2.47}$$

and

$$\frac{\partial V_g^r}{\partial \delta_{jf}} = -\boldsymbol{g}^T \sum_{i=j+1}^{n} \left(\frac{\partial \boldsymbol{W}_i}{\partial \delta_{jf}} \boldsymbol{M}_i^r\right). \tag{2.48}$$

The gravitational potential term in Lagrange's equations for the rigid proximal mass, equation (2.46), corresponds to the second term in equation (1.4) of Hollerbach's formulation of the rigid link dynamics.

The elastic potential for a slender link can be approximated by the strain energy in bending in the XY and XZ planes and twisting about the longitudinal, X, axis of the link. Along an incremental length $d\eta$ of link i, the elastic strain

energy is given by

$$dV_{ei} = \frac{1}{2}\left\{ E_i I_{xi} \left(\frac{\partial \theta_{xi}}{\partial \eta}\right)^2 + E_i I_{yi} \left(\frac{\partial \theta_{yi}}{\partial \eta}\right)^2 + G_i I_{zi} \left(\frac{\partial \theta_{zi}}{\partial \eta}\right)^2 \right\} d\eta, \qquad (2.49)$$

where for link i,

E_i is Young's modulus,

G_i is the shear modulus,

I_{yi} and I_{xi} are the second moments of area of the link cross section about the Y_i and Z_i axes respectively,

I_{zi} is the polar second moment of area of the link cross section, and

θ_{zi}, θ_{yi}, and θ_{xi} are the angles of rotation of the beam about the X_i, Y_i, and Z_i directions respectively at the generic point η along the length of the link.

Small deflections, that can be described by a summation of modal shapes and linear elasticity, have been assumed. With a truncated modal approximation for the link deflection, the angles of rotation of the beam, θ_{zi}, θ_{yi}, and θ_{xi}, at the generic point η are represented as a summation of modal amplitudes, δ_{ik}, times the angles of rotation, Θ_{zik}, Θ_{yik}, and Θ_{xik}, at the point η in the X_i, Y_i, and Z_i directions respectively due to the k_{th} mode of link i,

$$\begin{aligned} \theta_{zi} &= \sum_{k=1}^{m_i} \delta_{ik} \Theta_{zik}, \\ \theta_{yi} &= \sum_{k=1}^{m_i} \delta_{ik} \Theta_{yik}, \\ \theta_{xi} &= \sum_{k=1}^{m_i} \delta_{ik} \Theta_{xik}. \end{aligned} \qquad (2.50)$$

Integrated over the length of the link and summed over all the links the total elastic potential is given by

$$V_{ei} = \frac{1}{2}\int_0^{a_i} \left\{ E_i I_{xi} \left(\frac{\partial \theta_{xi}}{\partial \eta}\right)^2 + E_i I_{yi} \left(\frac{\partial \theta_{yi}}{\partial \eta}\right)^2 + G_i I_{zi} \left(\frac{\partial \theta_{zi}}{\partial \eta}\right)^2 \right\} d\eta. \qquad (2.51)$$

If the links are assumed to be uniform in cross section and if the modeshapes are orthogonal, the total strain energy for the manipulator may be written as

$$V_e = \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{m_i}\delta_{ik}^2\left\{E_iI_{xi}\int_0^{a_i}\left(\frac{\partial^2\Phi_{yik}}{\partial\eta^2}\right)d\eta + E_iI_{yi}\int_0^{a_i}\left(\frac{\partial^2\Phi_{zik}}{\partial\eta^2}\right)d\eta + G_iI_{xi}\int_0^{a_i}\left(\frac{\partial\Phi_{zik}}{\partial\eta}\right)d\eta\right\} \qquad (2.52)$$

where

Φ_{yik} and Φ_{zik} are the k_{th} modeshapes in bending in the XY and XZ planes of link i and

Φ_{xik} is the k_{th} modeshape in torsion about the X axis of link i.

The partial derivatives of the strain energy with respect to the joint variables, q_j, and the mode amplitudes, δ_{jf}, are

$$\frac{\partial V_e}{\partial q_j} = 0 \qquad (2.53)$$

and

$$\begin{aligned}\frac{\partial V_e}{\partial\delta_{jf}} &= \delta_{jf}\left\{E_jI_{xj}\int_0^{a_j}\left(\frac{\partial^2\Phi_{yjf}}{\partial\eta^2}\right)d\eta + E_jI_{yj}\int_0^{a_j}\left(\frac{\partial^2\Phi_{zjf}}{\partial\eta^2}\right)d\eta + G_iI_{xj}\int_0^{a_j}\left(\frac{\partial\Phi_{xjf}}{\partial\eta}\right)d\eta\right\}\\ &= \sum_{k=1}^{m_j}\delta_{jk}K_{jkf}, \qquad (2.54)\end{aligned}$$

where, for orthogonal modes, the elements of K_{jkf} are non-zero only for $k = f$.

2.3 Assembly of the Equations of Motion

Having derived expressions for the kinetic and potential energies of the manipulator and the derivatives of these required in Lagrange's equations, the equations of motion for joint j and modal coordinate jf may be written as

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_j}\right) - \frac{\partial T}{\partial q_j} + \frac{\partial V_e}{\partial q_j} + \frac{\partial V_g}{\partial q_j} = F_j \tag{2.55}$$

and

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\delta}_{jf}}\right) - \frac{\partial T}{\partial \delta_{jf}} + \frac{\partial V_e}{\partial \delta_{jf}} + \frac{\partial V_g}{\partial \delta_{jf}} = 0 \tag{2.56}$$

where F_j is the joint actuator torque.

After a considerable amount of work, the equations of motion for a serial link manipulator can be reduced to

$$I\ddot{z} = R \tag{2.57}$$

where I is an inertia matrix, z is a vector of the generalized coordinates,

$$z = (q_1\ \delta_{11}\ \delta_{12}\ \ldots\ \delta_{1m_1}\ \ldots q_2\ \delta_{21}\ \delta_{22}\ \ldots\ \delta_{2m_2}\ \ldots q_n\ \delta_{n1}\ \delta_{n2}\ \ldots\ \delta_{nm_n}),$$

and R is a vector of the remaining dynamics and external forcing terms.

2.3.1 Inertia Coefficients

To bring the equations of motion into the form of equation (2.57), the coefficients of the second derivatives of the generalized coordinates $\ddot{q}_j$ and $\ddot{\delta}_{jf}$ must be extracted from Lagrange's equations (2.55) and (2.56) to form the inertia matrix I. In order to separate the second derivatives of the joint variables $\ddot{q}_j$ and modal deflection amplitudes $\ddot{\delta}_{jf}$ from the expressions for $\bar{W}_i$ and $\hat{\bar{W}}_i$, it is necessary to define some intermediate transformations. Consider the product of transformations that compose W_i and $\hat{W}_i$

$$W_i = \underset{\hat{W}_{j-1}}{A_1 S_1 L_1 A_2 S_2 L_2 \cdots A_j} \underset{W_j}{S_j L_j \cdots A_{i-1} S_{i-1} L_{i-1} A_i} \overset{{}^j\tilde{W}_i}{\underset{{}^jW_i}{}}$$

and

$$W_i = \underset{\hat{W}_{j-1}}{A_1 S_1 L_1 A_2 S_2 L_2 \cdots A_j} \underset{W_j}{S_j L_j \cdots A_i S_i L_i}. \overset{{}^j\bar{W}_i}{\underset{{}^j\hat{W}_i}{}}$$

The transformation W_i may be written in two alternate forms

$$\begin{aligned} W_i &= \hat{W}_{j-1} A_j \, {}^j\tilde{W}_i \\ &= W_j^* L_j \, {}^jW_i. \end{aligned} \tag{2.58}$$

Similarly, the transformation $\hat{W}_i$ may be written as

$$\begin{aligned} \hat{W}_i &= \hat{W}_{j-1} A_j \, {}^j\bar{W}_i \\ &= W_j^* L_j \, {}^j\hat{W}_i. \end{aligned} \tag{2.59}$$

Differentiating equations (2.58) and (2.59) twice, the second derivatives $\tilde{W}_i$ and $\bar{\hat{W}}_i$ can be expressed as

$$\tilde{W}_i = \sum_{j=1}^{i} \hat{W}_{j-1} U_j \, {}^j\tilde{W}_i \, \tilde{q}_j + \sum_{j=1}^{i-1} \sum_{f=1}^{m_j} W_j^* M_{jf} \, {}^jW_i \, \tilde{\delta}_{jf} + \tilde{W}_{vi} \tag{2.60}$$

and

$$\bar{\hat{W}}_i = \sum_{j=1}^{i} \left(\hat{W}_{j-1} U_j \, {}^j\bar{W}_i \, \tilde{q}_j + \sum_{f=1}^{m_j} W_j^* M_{jf} \, {}^j\hat{W}_i \, \tilde{\delta}_{jf} \right) + \bar{\hat{W}}_{vi} \tag{2.61}$$

where

$$U_j = \frac{\partial A_j}{\partial q_j}.$$

The value of the transformations $\tilde{W}_{vi}$ and $\bar{\hat{W}}_{vi}$ can be calculated recursively as shown in equations (2.14) and (2.17) respectively, by eliminating the terms containing $\tilde{q}_j$ and $\tilde{\delta}_{jf}$. The result is

$$\tilde{W}_{vj} = \bar{\hat{W}}_{vj-1} A_j + 2\dot{\hat{W}}_{j-1} \frac{\partial A_j}{\partial q_j} \dot{q}_j + \hat{W}_{j-1} \frac{\partial^2 A_j}{\partial q_j^2} \dot{q}_j^2 \tag{2.62}$$

and

$$\dot{\tilde{W}}_{oj} = \tilde{W}_{oj} S_j L_j + 2\dot{W}_j^{\circ} \sum_{f=1}^{m_j} \dot{\delta}_{jf} M_{jf}. \tag{2.63}$$

Substituting the expressions for $\tilde{W}_i$ and $\dot{\tilde{W}}_i$ into Lagrange's equations and isolating the coefficients of the generalized coordinates allows the inertia matrix to be formed.

Isolating all occurrences of $\ddot{q}_h$ in joint equation j (2.55), the resulting coefficient of $\ddot{q}_h$ is

$$I_{jh} = 2\,Tr\left[\hat{W}_{j-1} U_j {}^{j}\tilde{F}_h U_h^T \hat{W}_{h-1}^T\right]$$

$$\text{for } j = 1 \cdots n, h = 1 \cdots n, \tag{2.64}$$

where

$${}^{j}\tilde{F}_h = \sum_{i=\max(h,j)}^{n} \left\{ {}^{j}\tilde{W}_i \left[S_i G_i S_i^T + J_{Ri} \right] {}^{h}\tilde{W}_i^T \right\}$$

$$\text{for } h = 1 \cdots n, j = 1 \cdots n. \tag{2.65}$$

The inertia coefficient of $\ddot{\delta}_{hk}$ in joint equation j (2.55) is I_{jhk}, the terms to be included depend on the values of j and h,

$$I_{jnk} = 2\,Tr\left[\hat{W}_{j-1} U_j {}^{j}\tilde{W}_n S_n D_{nk} S_n^T W_n^T\right]$$

$$\text{for } j = 1 \cdots n, h = n, k = 1 \cdots m_n, \tag{2.66}$$

$$I_{jhk} = 2\,Tr\left[\hat{W}_{j-1} U_j \left({}^{j}F_h M_{hk}^T W_h^{\circ T} + {}^{j}\tilde{W}_h S_h D_{hk} S_h^T W_h^T \right)\right]$$

$$\text{for } j = 1 \cdots n-1, h = j \cdots n-1, k = 1 \cdots m_h, \tag{2.67}$$

and

$$I_{jhk} = 2\,Tr\left[\hat{W}_{j-1} U_j {}^{j}F_h M_{hk}^T W_h^{\circ T}\right]$$

$$\text{for } j = 2 \cdots n, h = 1 \cdots j-1, k = 1 \cdots m_h, \tag{2.68}$$

where

$$ {}^{j}F_{h} = \sum_{i=\max(h+1,j)}^{n} \left\{ {}^{j}\tilde{W}_{i} \left[S_{i} G_{i} S_{i}^{T} + J_{Ri} \right] {}^{h}W_{i}^{T} \right\} $$

$$ \text{for } h = 1 \cdots n-1, j = 1 \cdots n. \qquad (2.69) $$

The coefficient of the $\tilde{q}_{h}$ in the deflection equation jf (2.56) is I_{jfh}, th terms to be included depend on the values of j and h,

$$ I_{nfh} = 2\,Tr \left[\hat{W}_{h-1} U_{h} {}^{h}\tilde{W}_{n} S_{n} D_{nf} S_{n}^{T} W_{n}^{T} \right] $$

$$ \text{for } j = n, h = 1 \cdots n, f = 1 \cdots m_{n}, \qquad (2.70) $$

$$ I_{jfh} = 2\,Tr \left[\hat{W}_{h-1} U_{h} \left({}^{h}F_{j} M_{jf}^{T} W_{j}^{*T} + {}^{h}\tilde{W}_{j} S_{j} D_{jf} S_{j}^{T} W_{j}^{T} \right) \right] $$

$$ \text{for } j = 1 \cdots n-1, h = 1 \cdots j, f = 1 \cdots m_{j}, \qquad (2.71) $$

and

$$ I_{jfh} = 2\,Tr \left[\hat{W}_{h-1} U_{h} {}^{h}F_{j} M_{jf}^{T} W_{j}^{*T} \right] $$

$$ \text{for } j = 1 \cdots n, h = j+1 \cdots n, f = 1 \cdots m_{j}. \qquad (2.72) $$

The coefficient of $\tilde{\delta}_{hk}$ in the deflection equation jf (2.56) is I_{jfhk}, the terms to be included depend on the values of j and h,

$$ I_{nfnk} = 2\,Tr \left[W_{n}^{*} C_{nkf} W_{n}^{*T} \right] $$

$$ \text{for } j = h = n, f = 1 \cdots m_{n}, k = 1 \cdots m_{n}, \qquad (2.73) $$

$$ I_{jfjk} = 2\,Tr \left[W_{j}^{*} M_{jf} {}^{j}N_{j} M_{jk}^{T} W_{j}^{*T} + W_{j}^{*} C_{jkf} W_{j}^{*T} \right] $$

$$ \text{for } j = h = 1 \cdots n-1, f = 1 \cdots m_{j}, k = 1 \cdots m_{j}, \qquad (2.74) $$

$$ I_{jfnk} = 2\,Tr \left[W_{j}^{*} M_{jf} {}^{j}W_{n} S_{n} D_{nk} S_{n}^{T} W_{n}^{T} \right] $$

$$ \text{for } j = 1 \cdots n-1, h = n, f = 1 \cdots m_{j}, k = 1 \cdots m_{n}, \qquad (2.75) $$

and

$$I_{jfhk} = 2\,Tr\left[W_j^* M_{jf}\left({}^jN_h M_{hk}^T W_h^{*T} + {}^jW_h S_h D_{hk} S_h^T W_h^T\right)\right]$$
$$\text{for } j = 1\cdots n-1, h = j+1\cdots n-1, f = 1\cdots m_j, k = 1\cdots m_h, \qquad (2.76)$$

where

$${}^jN_h = \sum_{i=\max(h+1,j+1)}^{n}\left\{{}^jW_i\left[S_i G_i S_i^T + J_{Ri}\right]{}^hW_i^T\right\}$$
$$\text{for } h = 1\cdots n-1, j = 1\cdots n-1. \qquad (2.77)$$

The values of the ${}^j\tilde{F}_h$, ${}^jF_h$, and ${}^jN_h$ arrays are computed as follows. Starting atg the end of the manipulator with

$$\begin{aligned} {}^n\tilde{F}_n &= S_n G_n S_n^T + J_{Rn} \\ &= G_n^*, \end{aligned} \qquad (2.78)$$

the values of ${}^j\tilde{F}_h$ are computed recursively working back to the base, varying j faster than h,

$${}^j\tilde{F}_h = \begin{cases} G_n^* + {}^j\tilde{F}_{j+1}\left(S_j L_j A_{j+1}\right)^T, & \text{for } j = h \neq n \\ S_j L_j A_{j+1}\,{}^{j+1}\tilde{F}_h, & \text{for } 1 \leq j < h. \end{cases} \qquad (2.79)$$

Then, the ${}^jF_h$ and ${}^jN_h$ are computed

$${}^jF_h = {}^j\tilde{F}_{h+1} A_{h+1}^T \qquad (2.80)$$

and

$${}^jN_h = A_{j+1}\,{}^{j+1}F_h. \qquad (2.81)$$

2.3.2 Remaining Dynamics Terms

The remaining dynamics vector R is composed of those terms in Lagrange's equations remaining after the removal of the coefficients of the second derivatives

of the generalized coordinates. Excluding the coefficients of $\ddot{q}_h$ and $\ddot{\delta}_{hk}$ from equation (2.55), the remaining dynamics of joint j can be written as

$$\begin{aligned} R_j &= -2Tr\sum_{i=j}^{n}\left[\hat{W}_{j-1}U_j{}^j\tilde{W}_i\left(G_i^*\tilde{W}_{vi}^T + 2\sum_{k=1}^{m_i}\dot{\delta}_{ik}S_iD_{ik}S_i^T\dot{W}_i^T\right)\right] \\ &\quad +g^T\sum_{i=j}^{n}\left[\hat{W}_{j-1}U_j{}^j\tilde{W}_i\left(S_ir_i + M_i^r\right)\right] \\ &\quad +F_j \\ &= -2Tr\left[\hat{W}_{j-1}U_jQ_j\right] + g^T\hat{W}_{j-1}U_jP_j + F_j \end{aligned} \tag{2.82}$$

where the values of Q_j and P_j are computed recursively, starting with the last link and working back to the base,

$$Q_j = \begin{cases} G_n^*\tilde{\bar{W}}_{vn}^T + 2\sum_{k=1}^{m_n}\dot{\delta}_{nk}S_nD_{nk}S_n^T\dot{W}_n^T & \text{for } j = n \\ G_j^*\tilde{\bar{W}}_{vj}^T + 2\sum_{k=1}^{m_j}\dot{\delta}_{jk}S_jD_{jk}S_j^T\dot{W}_j^T + S_jL_jA_{j+1}Q_{j+1}, & \text{for } 1 \le j < n, \end{cases} \tag{2.83}$$

and

$$P_j = \begin{cases} S_nr_n + M_n^r & \text{for } j = n \\ S_jr_j + M_j^r + S_jL_jA_{j+1}P_{j+1}, & \text{for } 1 \le j < n. \end{cases} \tag{2.84}$$

Excluding the coefficients of $\ddot{q}_h$ and $\ddot{\delta}_{hk}$ from equation (2.56), the remaining dynamics of the deflection equation jf can be written as

$$
\begin{aligned}
R_{jf} &= -2\,Tr \sum_{i=j+1}^{n} \left[W_{j-1}^{*} M_{jf}\, {}^{j}W_i \left(G_i^{*} \tilde{W}_{vi}^{T} + 2 \sum_{k=1}^{m_i} \dot{\delta}_{ik} S_i D_{ik} S_i^{T} \dot{W}_i^{T} \right) \right] \\
&\quad -2\,Tr \left[\tilde{W}_{vj}^{T} D_{jf} W_j^{*T} + 2\, \dot{W}_j^{*} \sum_{k=1}^{m_j} \dot{\delta}_{jk} C_{jkf} W_j^{*T} \right] \\
&\quad + g^{T} \sum_{i=j+1}^{n} \left[W_j^{*} M_{jf}\, {}^{j}W_i \left(S_i r_i + M_i^{r} \right) \right] + g^{T} W_j^{*} e_{jf} \\
&\quad - \sum_{k=1}^{m_j} \delta_{jk} K_{jkf} \\
&= \begin{cases}
-2\,Tr \left[\tilde{W}_{vn} S_n D_{nf} S_n^{T} W_n^{T} + 2\, \dot{W}_n^{*} \sum_{k=1}^{m_n} \dot{\delta}_{nk} C_{nkf} W_n^{*T} \right] & \\
\quad - \sum_{k=1}^{m_n} \delta_{nk} K_{nkf} + g^{T} W_n^{*} e_{nf}, & \text{for } j = n \\
-2\,Tr \left[W_j^{*} M_{jf} A_{j+1} Q_{j+1} \right] & \\
\quad -2\,Tr \left[\tilde{W}_{vj} S_j D_{jf} S_j^{T} W_j^{T} + 2\, \dot{W}_j^{*} \sum_{k=1}^{m_j} \dot{\delta}_{jk} C_{jkf} W_j^{*T} \right] & \\
\quad - \sum_{k=1}^{m_j} \delta_{jk} K_{jkf} + g^{T} W_j^{*} \left(M_{jf} A_{j+1} P_{j+1} + e_{jf} \right), & \text{for } 1 \le j < n.
\end{cases}
\end{aligned}
\tag{2.85}
$$

The equations of motion can now be put in the form of equation (2.57). This equation can then be solved for the vector of generalized accelerations and integrated to produce the time history of the generalized coordinates.

Chapter 3

Model Implementation and Verification

In this chapter, a numerical simulation of a flexible link manipulator, called FLX, based on Book's [27] algorithm is described and some attempts at verifying the correctness of the simulation are presented.

3.1 Overview of FLX

A numerical simulation of a flexible link manipulator has been developed utilizing Book's [27] algorithm, as presented in Chapter 2, to model the flexible link dynamics. A flow chart of the simulation is shown in Figure 3.1. If the manipulator is driven, the simulation accepts as input the cubic spline coefficients of a joint space trajectory. Independent fixed gain PID (proportional - integral - derivative) joint controllers supply joint torques based on the error between the command and actual joint positions. The actuator dynamics are not modelled,

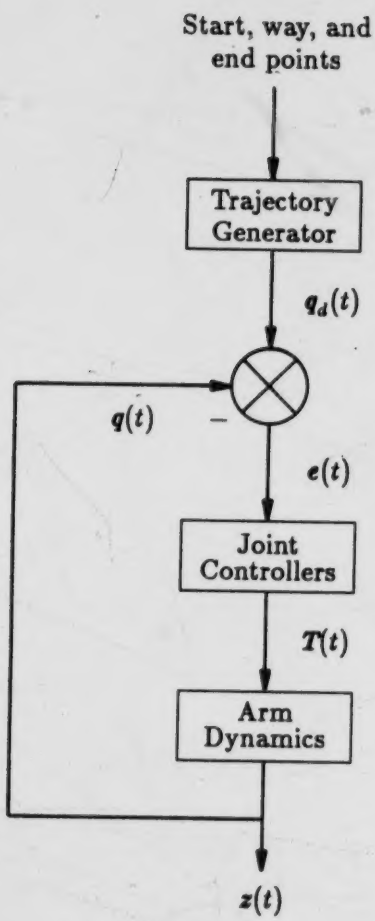

Figure 3.1: A flowchart of the FLX simulation

however a lumped mass and inertia at the joint can be used to simulate the load of a joint motor on the manipulator. If the manipulator is undriven, the input to the simulation is an initial kinematic state.

Simulation of the response of a manipulator to a set of prescribed control inputs requires repeated solution of the forward dynamics, involving integration of the equations of motion. The equations of motion (2.55) developed in Chapter 2 are in inverse dynamic form. Given the kinematic state of the manipulator, they can be used to calculate the joint torques required to produce that motion. These equations must be inverted to solve for the generalized accelerations,

$$\ddot{z} = I^{-1}R. \tag{3.1}$$

These equations are then integrated to produce a time history of the state $\{z, \dot{z}\}$ of the system, given an initial state $\{z, \dot{z}\}_0$ and a prescribed set of joint torques. Because of the complexity of the inertia matrix, the inversion is performed numerically using Cholesky decomposition. The use of Cholesky decomposition, a form of LU decomposition, takes advantage of the symmetry of the inertia matrix. A fourth order Runge-Kutta scheme was used to perform the integration. The complete dynamics equation 3.1 was evaluated at each of the four function evaluations required per time step in the fourth order Runge-Kutta algorithm. Other integration schemes were not investigated.

3.2 Limitations and assumptions

FLX models a serial chain of links connected by actuated revolute joints, prismatic joints can not be modelled. The links must be uniform in cross section and may be rigid or flexible. The flexure of a link is described by a truncated

series of assumed normal modes [31,33],

$$v_i(\eta,t) = \sum_{i=1}^{m_i} \delta_{ij}(t)\Phi_{ij}(\eta) \tag{3.2}$$

where

$v_i(\eta,t)$ is the displacement at time t of a generic point a distance η along the neutral axis of link i,

$\Phi_{ij}(\eta)$ is the shape of the j_{th} mode of the deflection of link i, and

$\delta_{ij}(t)$ is its time varying amplitude.

The modeshapes chosen are eigenfunctions of the Bernoulli - Euler flexure equation

$$EI\frac{\partial^4 v_i}{\partial \eta^4} + m\frac{\partial^2 v_i}{\partial t^2} = 0 \tag{3.3}$$

with boundary conditions appropriate for a fixed free beam,

$$v_i(0,t) = 0,$$

$$\frac{\partial v_i}{\partial \eta}(0,t) = 0,$$

$$\frac{\partial^2 v_i}{\partial \eta^2}(a,t) = 0,$$

and

$$\frac{\partial^3 v_i}{\partial \eta^3}(a,t) = 0$$

where

EI is the stiffness of the beam,

m is the mass per unit length of the beam, and

a is the length of the beam.

Thus, the flexible links used in an FLX simulation must not violate the assumptions of Bernoulli - Euler beam theory:

1. the beam is initially straight,
2. the depth of the beam is small in comparison to the radius of curvature at maximum flexure,
3. plane sections remain plane at all phases of an oscillation,
4. the beam is free from axial load,
5. rotatory inertia can be ignored, and
6. shear deformation can be neglected.

Since the radius of curvature of a beam at maximum flexure will be smaller for the higher modes than for the lower modes, inclusion of higher modes may violate the assumptions of Bernoulli - Euler beam theory. Accordingly, the flexible links used in FLX should only include the first few modes. The modeshapes of a fixed free beam are

$$\Phi_{ij}(\eta) = \left\{\frac{\sinh(\alpha_j a_i) + \sin(\alpha_j a_i)}{\cosh(\alpha_j a_i) + \cos(\alpha_j a_i)}\right\}\{\cosh(\alpha_j \eta) - \cos(\alpha_j \eta)\} - \{\sinh(\alpha_j \eta) - \sin(\alpha_j \eta)\} \tag{3.4}$$

where $\alpha_j a_i$ is the j_{th} root of the frequency equation

$$\cos(\alpha_j a_i)\cosh(\alpha_j a_i) + 1 = 0$$

associated with equation (3.3) subject to the fixed free boundary conditions for link i. The first three modeshapes and their natural frequencies are shown in Figure 3.2. The modal coordinates for the fundamental modes of flexure are shown in Figure 3.3.

There must be a minimum of two links in an FLX model, because of the dimensioning of arrays in FLX. However, joints can be locked, effectively removing a link from the computation. Also, due to the coding of FLX, a flexible link in

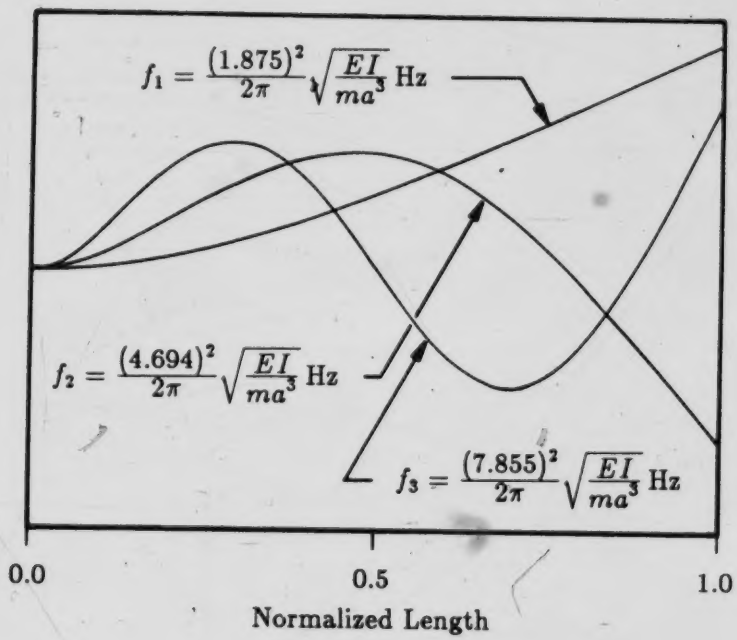

Figure 3.2: The first three modeshapes of a fixed free beam

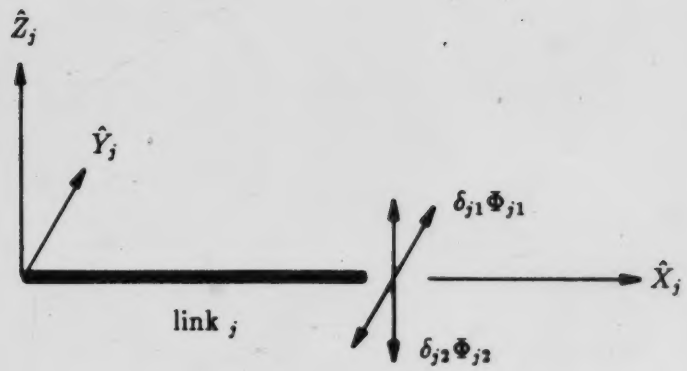

Figure 3.3: The flexural modal coordinates of the FLX link

FLX must have at least one mode of flexure in each of the $\hat{X}\hat{Y}$ and $\hat{X}\hat{Z}$ planes. This is not a restriction, since a link can be made rigid in one of the $\hat{X}\hat{Y}$ or $\hat{X}\hat{Z}$ bending planes, independent of the link stiffness in the other bending plane by choosing a sufficiently large stiffness in that plane.

With the exception of the terminal link n in a serial chain of links, the choice of modeshapes of a fixed free beam to describe the flexure of any other link i in the chain is clearly incorrect. The boundary condition at the distal end of link i, $i < n$, is neither free nor fixed. There are moment and shear at the distal end of link i due to the subsequent links in the chain and the load mass. Further, the boundary condition at the distal end of link i changes with time as the configuration of the manipulator is a function of time. However, for the purposes of verifying the correctness of the coding of FLX, the fixed free boundary conditions are applied to all of the links in the chain.

The inclusion of torsional modes of oscillation about the longitudinal $\hat{X}$ axis of a link in FLX is optional.

3.3 Model verification

Without comparison to a physical model or perhaps a finite element model, verification of the numerical model and solution technique presented herein is difficult. However, checks can be made, which although not conclusive, do provide a strong indication of the correctness of the numerical model. First of all, the model must agree with simpler models for which accepted solutions exist. Secondly, the numerical model must obey physical conservation laws. In the absence of external forcing, the numerical model must conserve energy. And thirdly, the numerical model must demonstrate closure:

$$\text{Joint Torques} \underset{\text{Inverse Dynamics}}{\overset{\text{Forward Dynamics}}{\rightleftharpoons}} \text{Kinematic State.}$$

If a joint torque history is applied to a manipulator model, the joint accelerations, calculated using the direct dynamics, can be integrated to follow the kinematic state of the manipulator in time. The time history of the kinematic state of the manipulator can then be input into the inverse dynamics to calculate the joint torques required to produce the motion. To demonstrate closure the control joint torque history applied to the manipulator and the joint torque history required to produce the resulting motion must be identical [1].

A number of test cases used to verify the numerical model and solution technique are presented. For each of the test cases, the equations of motion were derived using a scalar Lagrangian formulation. The equations of motion were then integrated using the same fourth order Runge-Kutta scheme used in the implementation of Book's [27] algorithm.

For the test cases that follow the flexible link used was a thin walled tube annular in cross section and 4m in length. The properties of the link are given in Table 3.1. The choice of the link parameters was dictated by integration timestep considerations. Since the natural frequency of flexural oscillation of a beam is directly proportional to the stiffness and inversely proportional to the length of the beam, increasing the length of the link and decreasing the stiffness decreases the frequency of lateral vibration. The choice of a long slender link allows a larger timestep to be used than would be possible for a shorter, stiffer link, thus reducing the time required for the computation. In the test cases that follow, only the fundamental mode of flexural vibration was used and, although the numerical model can accomodate link torsion, torsion was ignored.

Flexible Link Parameters	
mass, m	10.955 kg
outside diameter	6.00×10^{-2} m
inside diameter	5.62×10^{-2} m
length, a	4.0 m
centroidal second moments of area	
I_{yi} and I_{zi}	1.476×10^{-7} m^4
I_{xi}	2.952×10^{-7} m^4
Young's modulus, E	2.0×10^{11} Pa
density, ρ	7832 kg/m^3

Table 3.1: Properties of the flexible link

3.3.1 Single rigid link compound pendulum

The first test case presented is a rigid compound pendulum (SRPEN), shown in Figure 3.4. The equation of motion of a rigid compound pendulum of uniform link cross section and length a is

$$\ddot{\phi} + \frac{3g}{2a}\sin\phi = 0 \tag{3.5}$$

where ϕ is the angular displacement of the link from vertical. The rigid compound pendulum is a useful test case, since an exact elliptic integral solution exists for the period τ of the rigid body motion,

$$\tau = \sqrt{\frac{32a}{3g}} \int_0^{\frac{\pi}{2}} \frac{d\gamma}{\sqrt{1 - k^2 \sin^2\gamma}} \tag{3.6}$$

where

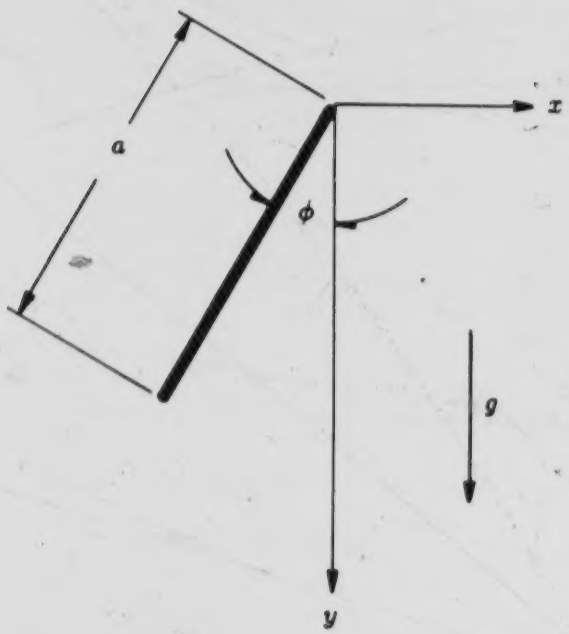

Figure 3.4: SRPEN geometry used to simulate the single rigid link compound pendulum

$k = \sin \frac{\phi_{max}}{2}$ and

γ is a dummy variable of integration

and for small displacements where $\sin\phi \simeq \phi$, a constant analytical expression exists for the period, τ, of the rigid body motion,

$$\tau = 2\pi\sqrt{\frac{2a}{3g}}. \tag{3.7}$$

The FLX geometry used to simulate a rigid compound is shown in Figure 3.5. Two links are required to model the rigid link compound pendulum in FLX because of the dimensioning of arrays in FLX. Link 1 is not free to rotate about $\hat{Z}_0$, that is, joint 1 is locked, and link 2 is free to rotate about $\hat{Z}_1$. The gravitational force on the link is in the $-\hat{X}_0$ direction.

The response of the compound pendulum, 4 m in length, simulated using FLX with an integration time step of 1 ms, to an initial displacement of 0.4 rads is shown in Figure 3.6. The period of the motion, $\tau = 3.308 \pm 0.001$ s, is close to the elliptic integral value of 3.309 s and analytical value of 3.276 s for small displacements. By decreasing the magnitude of the initial displacement from 0.4 rads to 0.05 rads, the period of oscillation approaches the analytical value as shown in Table 3.2. For an initial displacement of 0.05 rads the period is in agreement with the analytical value for small displacements. SRPEN and FLX give identical values for the period of the motion. The solutions using both SRPEN and FLX are conservative, the total system energy remains constant to within eight decimal places at its initial value.

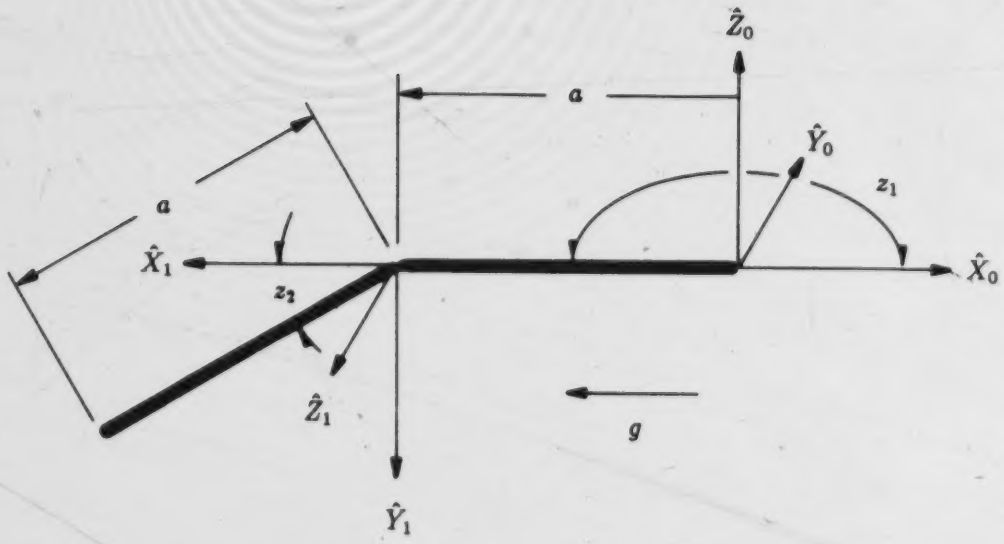

Figure 3.5: FLX geometry used to simulate the single rigid link compound pendulum

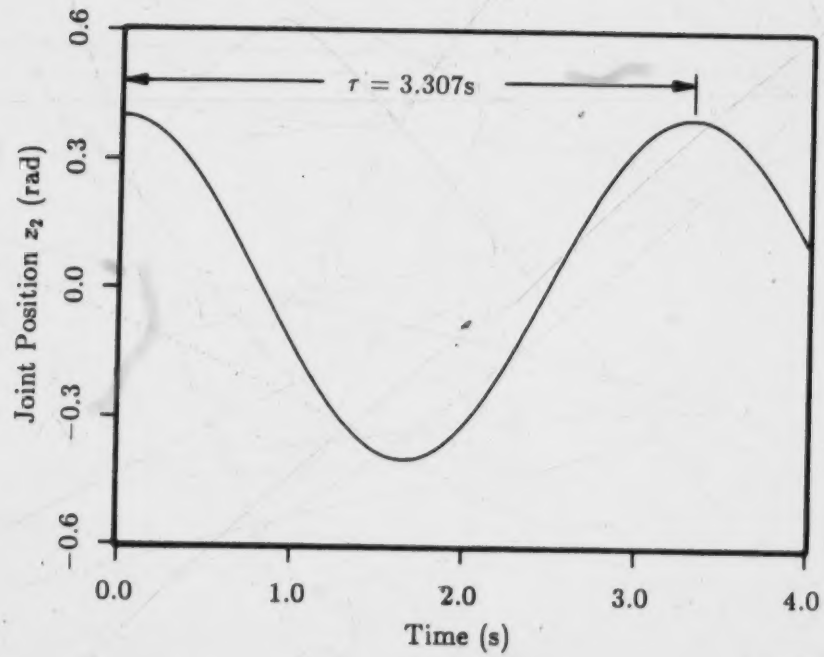

Figure 3.6: Response of a single rigid link compound pendulum to an initial displacement

Initial Displacement ϕ(rad)	Period τ(s)		
	SRPEN	FLX	Elliptic Integral
0.05	3.276	3.276	3.276
0.1	3.278	3.278	3.278
0.2	3.284	3.284	3.284
0.4	3.308	3.308	3.309

Table 3.2: Effect of the initial displacement on the period of oscillation of the rigid compound pendulum

3.3.2 Double rigid link compound pendulum

The second test case presented is a double rigid link compound pendulum (DRPEN), shown in Figure 3.7. Both links are identical and of length a; the joint axes are parallel. The equations of motion of the double link compound pendulum are

$$(5 + 3\cos\phi_2)\ddot{\phi}_1 + (1 + \frac{3}{2}\cos\phi_2)\ddot{\phi}_2 - \frac{3}{2}\sin\phi_2(\dot{\phi}_1 + \dot{\phi}_2)\dot{\phi}_2 + \frac{6g}{a}\sin\phi_1 = 0 \quad (3.8)$$

and

$$(1 + \frac{3}{2}\cos\phi_2)\ddot{\phi}_1 + \ddot{\phi}_2 + \frac{3}{2}\sin\phi_2(\dot{\phi}_1)^2 + \frac{3g}{2a}\sin\phi_2 = 0 \quad (3.9)$$

where

ϕ_1 is the angular displacement of the base link from vertical and

ϕ_2 is the angular displacement of the outer link from the longitudinal axis of the base link.

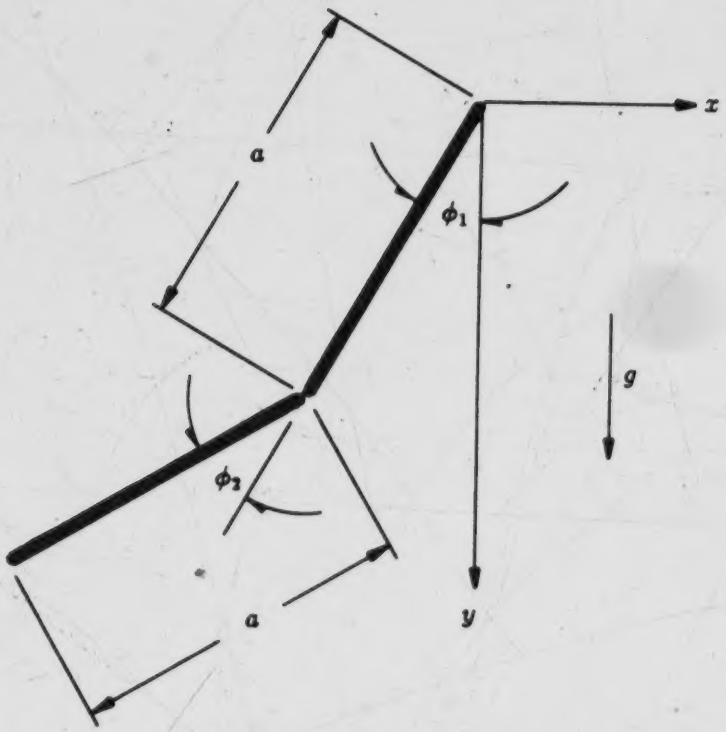

Figure 3.7: DRPEN geometry used to simulate the double link rigid compound pendulum

The FLX geometry used to model the double link compound pendulum is shown in Figure 3.8. The joint axes are parallel and both joints are free to move. The gravitiational force on the system is in the $-\hat{X}_0$ direction.

The response of the system, simulated using DRPEN with an integration time step of 1 ms, to an initial displacement of 0.4 rads of both links is shown in Figure 3.9. The response of the system, simulated using FLX with an integration time step of 1 ms, to the same initial conditions is shown in Figure 3.10. The solutions produced by DRPEN and FLX are identical. Also both solutions are conservative, the total system energy remains constant within eight decimal places at its initial value.

3.3.3 Single flexible link compound pendulum

The third test case presented is the flexible compound pendulum (FPEN), shown in Figure 3.11. The link has length a, mass m, and stiffness EI. The deflection of the link from its undeformed position is

$$v(\eta,t) = \varsigma(t)\Phi(\eta) \tag{3.10}$$

where $\varsigma(t)$ is the time varying amplitude or modal coordinate of the single modeshape $\Phi(\eta)$ used to describe the deformed shape of the link. The modeshape used to describe the flexibility of the link is the fundamental modeshape of a fixed free beam. The equations of motion of the flexible pendulum are

$$\left(\frac{ma^2}{3} + mF\varsigma^2\right)\ddot{\phi} - mGa\ddot{\varsigma} + 2mF\varsigma\dot{\varsigma}\dot{\phi} + mg\left(\frac{a}{2}\sin\phi - H\varsigma\cos\phi\right) = T \tag{3.11}$$

$$-Ga\ddot{\phi} + mF\ddot{\varsigma} + \left(\frac{KEI}{4a^3} - mF\dot{\varsigma}^2\right)\varsigma - mg\sin\phi = 0 \tag{3.12}$$

where

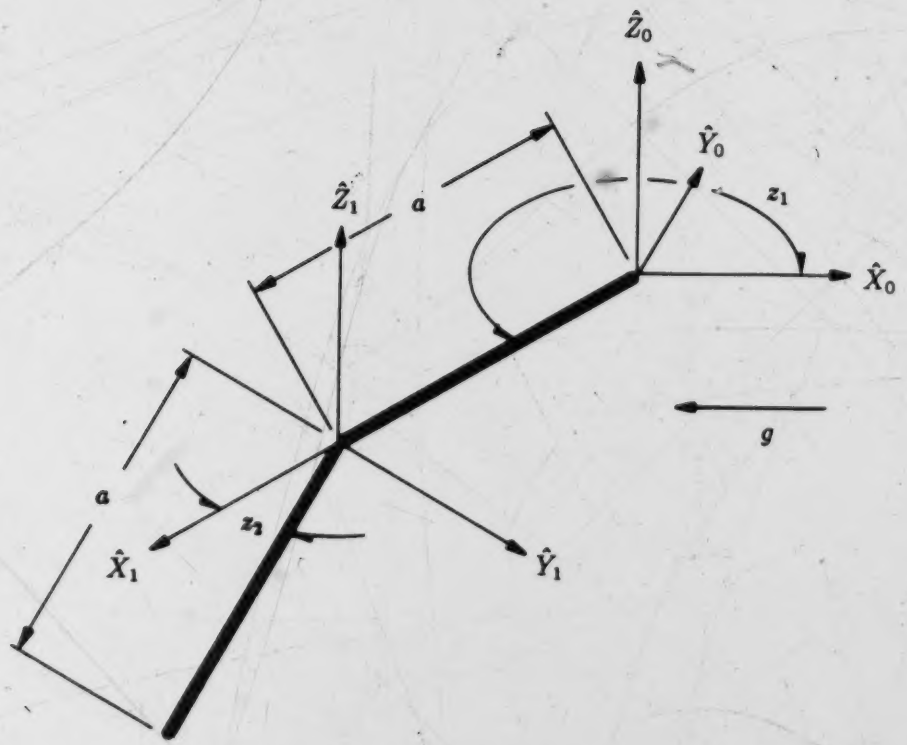

Figure 3.8: FLX geometry used to simulate the double link rigid compound pendulum

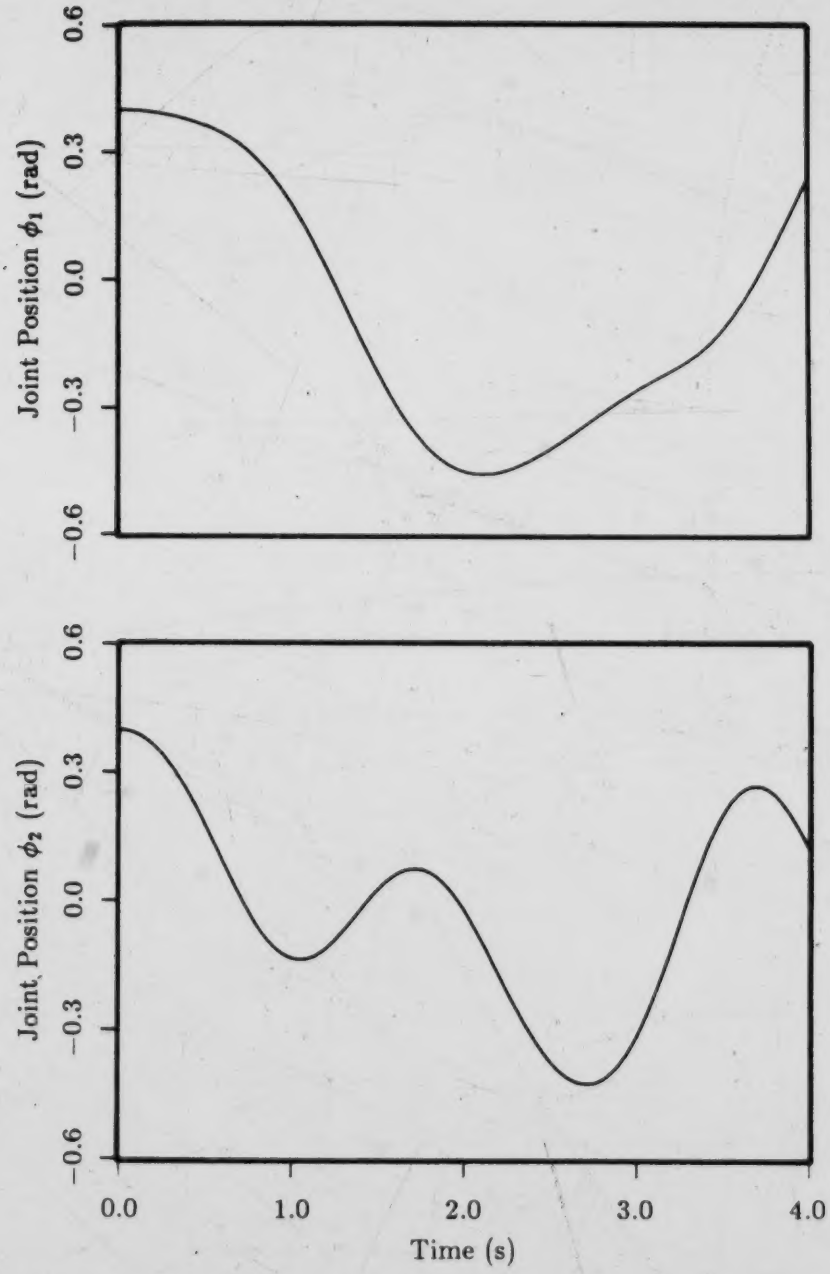

Figure 3.9: Response of a double rigid link compound pendulum to an initial displacement, simulated using DRPEN with a 1 ms timestep

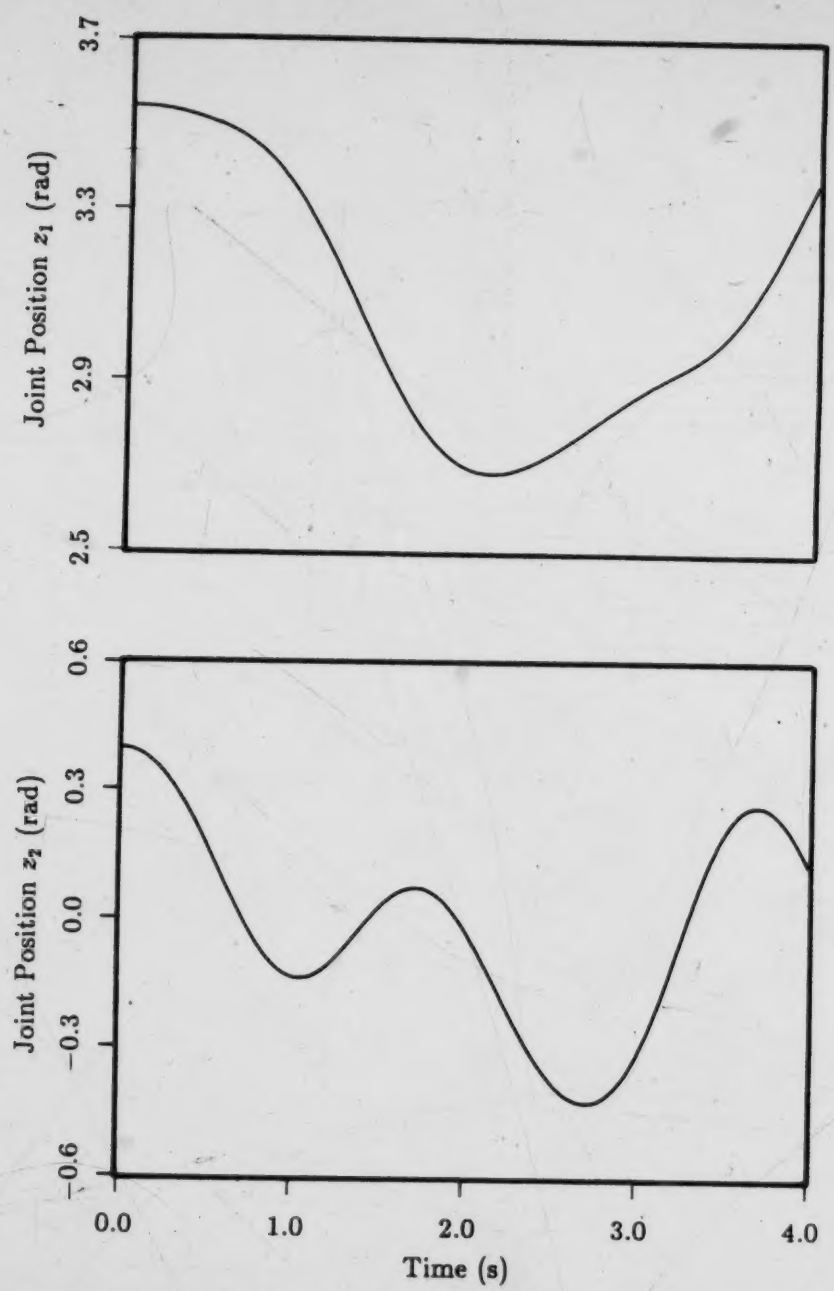

Figure 3.10: Response of a double rigid link compound pendulum to an initial displacement, simulated using FLX with a 1 ms timestep

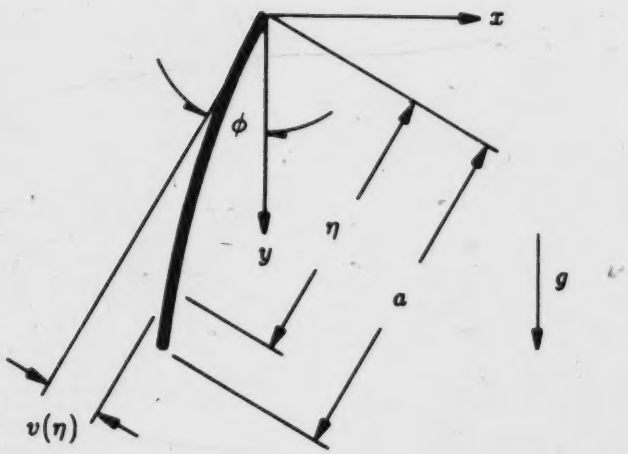

Figure 3.11: FPEN geometry used to model the single flexible link compound pendulum

ϕ is the angular displacement of the undeformed link from vertical,

ς is the modal coordinate describing the deflection of the link,

T is an applied joint torque, and

$$F = \frac{1}{a}\int_0^a V^2(\eta)d\eta \tag{3.13}$$

$$G = \frac{1}{a^2}\int_0^a \eta V(\eta)d\eta \tag{3.14}$$

$$H = \frac{1}{a}\int_0^a V(\eta)d\eta \tag{3.15}$$

$$K = (\alpha a)^4 V^2(a) \tag{3.16}$$

and where αa is the first root of the frequency equation

$$\cos(\alpha a)\cosh(\alpha a) + 1 = 0$$

associated with equation (3.3) subject to the fixed free boundary conditions. The FLX geometry used to model the flexible pendulum is shown in Figure 3.12. This geometry is essentially the same as that used to model the rigid pendulum, except for the inclusion of the single mode to model the flexure of the link.

The response of the flexible pendulum, simulated using FPEN using a 10 ms timestep, to an initial displacement of 0.4 rads and a modal coordinate $\varsigma = 0.012$, corresponding to an endpoint deflection of 0.0327 m, is shown in Figure 3.13. The joint angle oscillates with a period $\tau = 3.31 \pm 0.01$ s, close to the period of motion of the rigid link compound pendulum $\tau = 3.307 \pm 0.001$ s with the same initial displacement. Superimposed on the rigid body motion is a higher frequency oscillation, $\tau = 50 \pm 10$ ms, due to the flexure of the link. The amplitude of the oscillation of the link is seen to decay with time and this decay is reflected in a decay of the total system energy with time. The decay of the total system energy

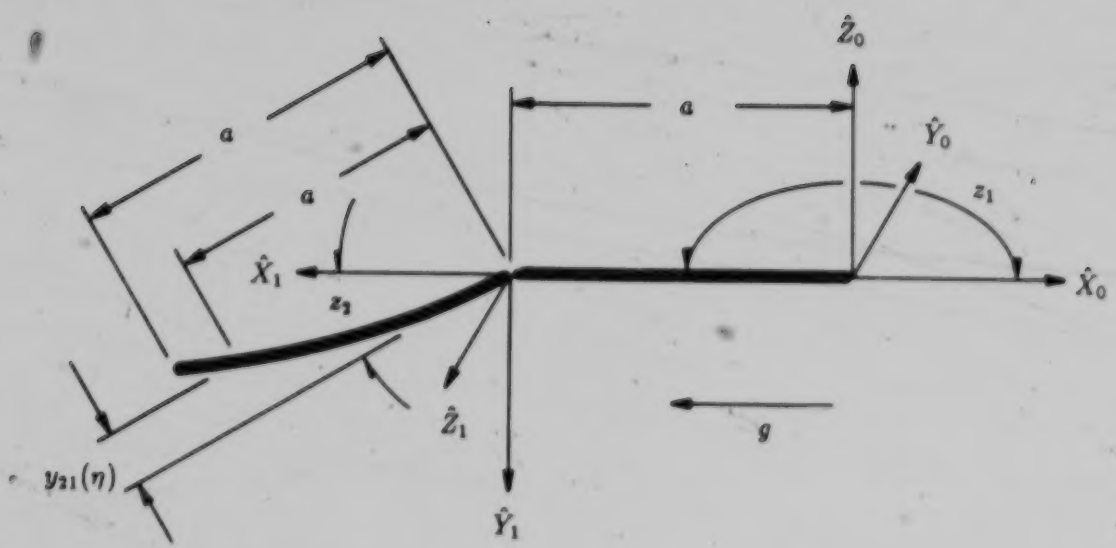

Figure 3.12: FLX geometry used to model the single flexible link compound pendulum

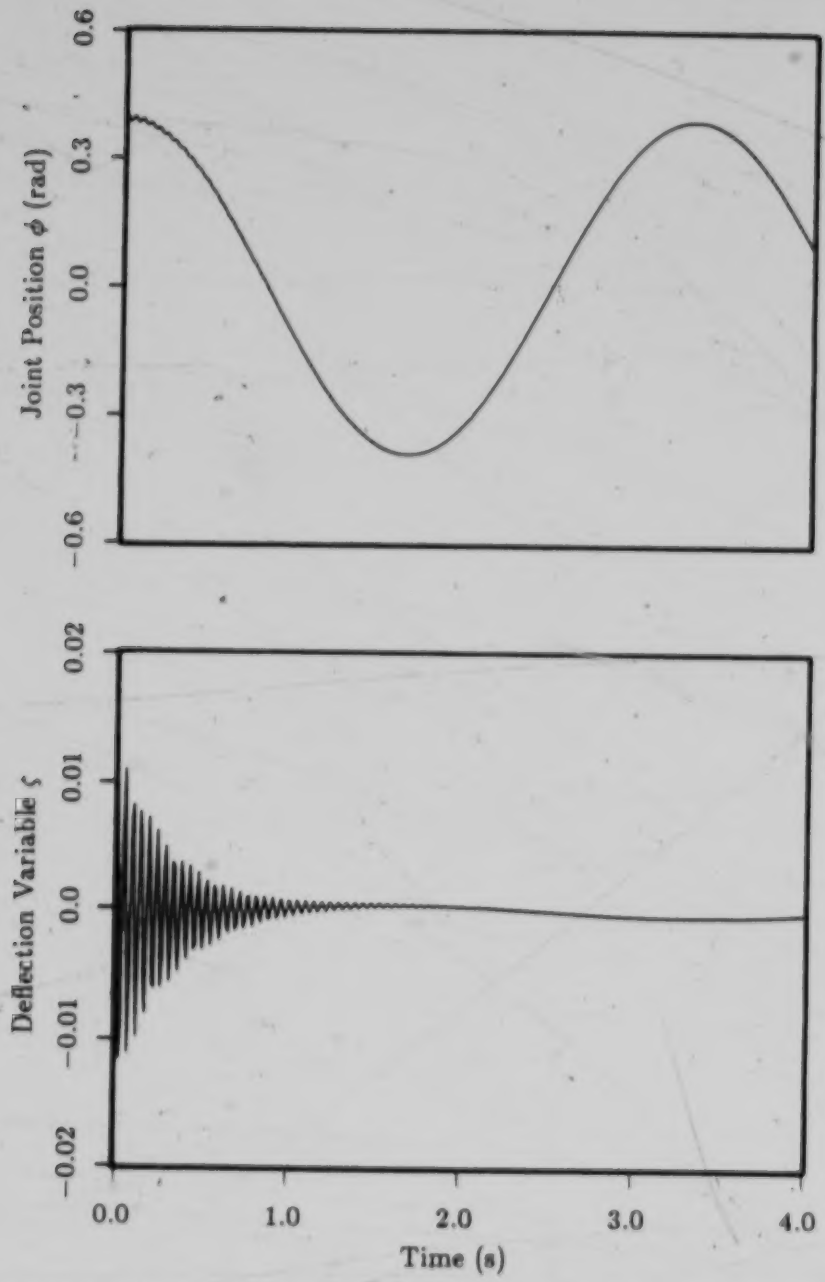

Figure 3.13: Response of a flexible link compound pendulum to an initial displacement and deflection, simulated using FPEN with a 10 ms timestep

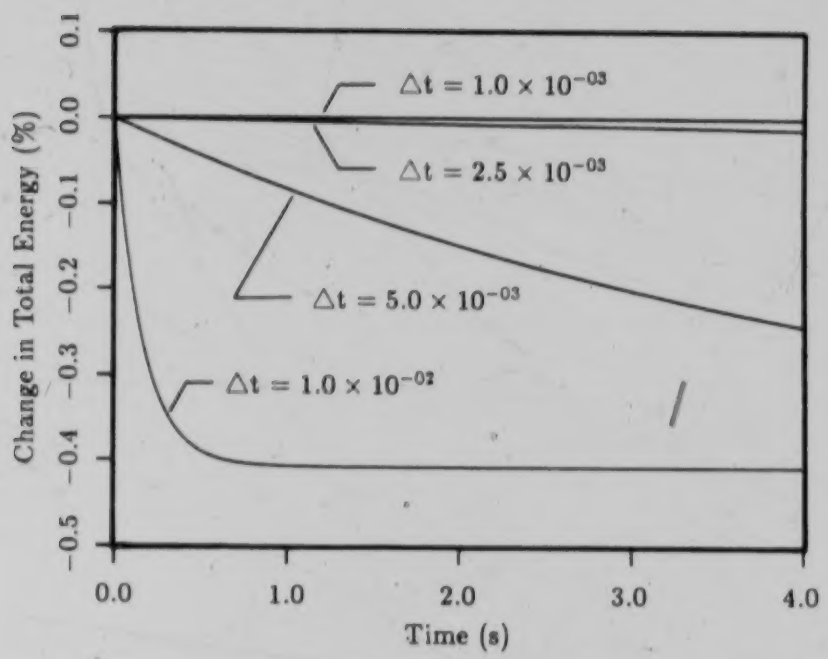

Figure 3.14: The effect of varying integration time step on total system energy decay

with varying integration time step is shown in Figure 3.14. Evidently, the numerical damping of the flexural oscillation is a function of the integration timestep. With a time step 10 ms, the fourth order Runge-Kutta scheme is incapable of following the oscillation in the flexure of the link. The amplitude of the flexural oscillation has fallen to approximately zero by 2 s, corresponding to the plateau in the total system energy. The quasi-static deflection of the link in response to the joint motion is all that remains after approximately 2 s into the simulation. Reduction of the time step to 1 ms, approximately $\frac{1}{50}$th of the period of the link flexural oscillation 47 ± 1 ms, results in a negligible decrease in total system energy over the duration of the simulation.

The response of the flexible pendulum to the same initial conditions using FPEN and FLX with a 1 ms timestep, are shown in Figure 3.15 and Figure 3.16, respectively. Both solutions are conservative, the total system energy remains constant to within eight decimal places. The period of the joint motion, 3.307 ± 0.001 s, produced by both FPEN and FLX, agrees with the period of the joint motion of the rigid link compound pendulum. The period of the link flexural oscillation, 47 ± 1 ms, is also the same for both FPEN and FLX. The period of the link flexural oscillation predicted by FPEN and FLX do not agree with the analytical value of the fundamental period, 275 ms of a fixed free beam. However, the flexure of the link is observed to back drive the joint as shown in Figures 3.13, 3.15, and 3.16, that is the base of the link is free to rotate in response to the flexure of the link. Thus, since the fixed end condition at the base of the link is violated, this disagreement is not unexpected.

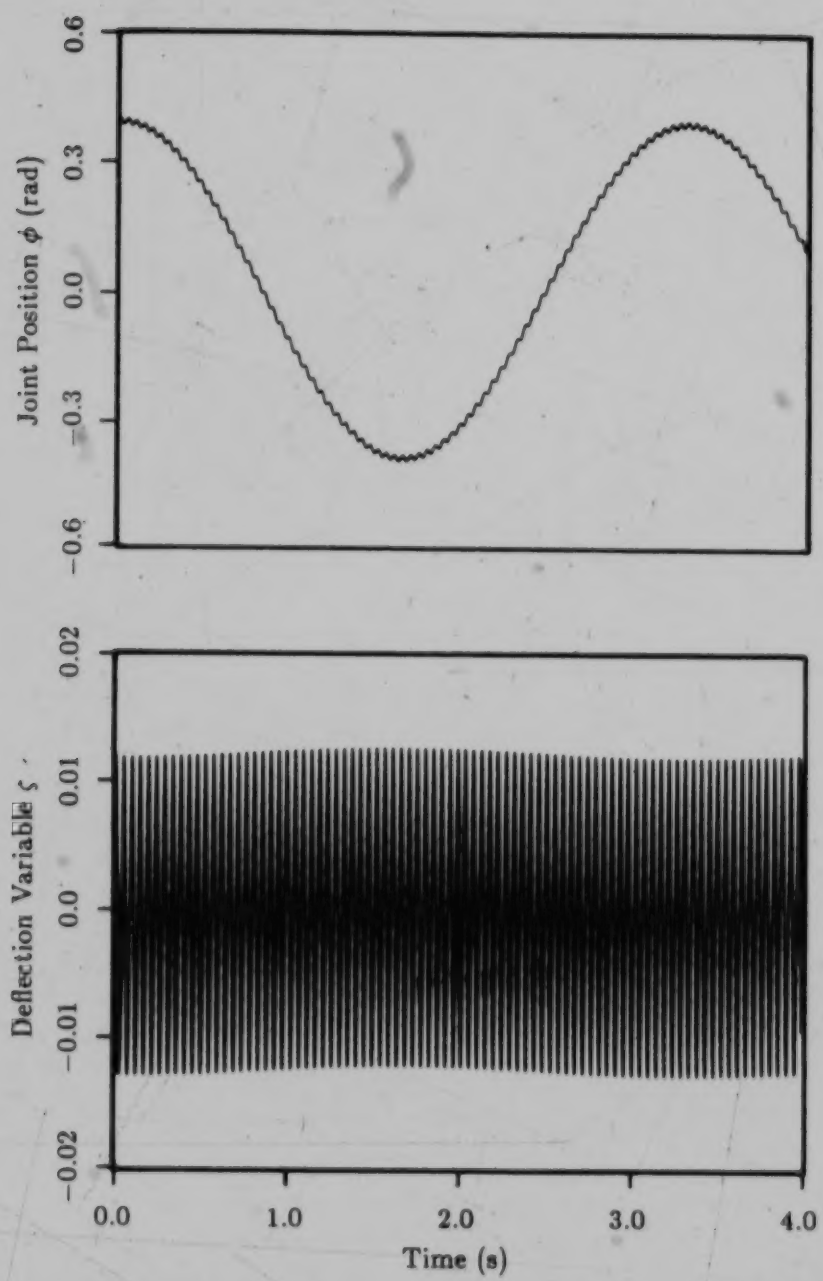

Figure 3.15: Response of a flexible link compound pendulum to an initial displacement and deflection, simulated using FPEN with a 1 ms timestep

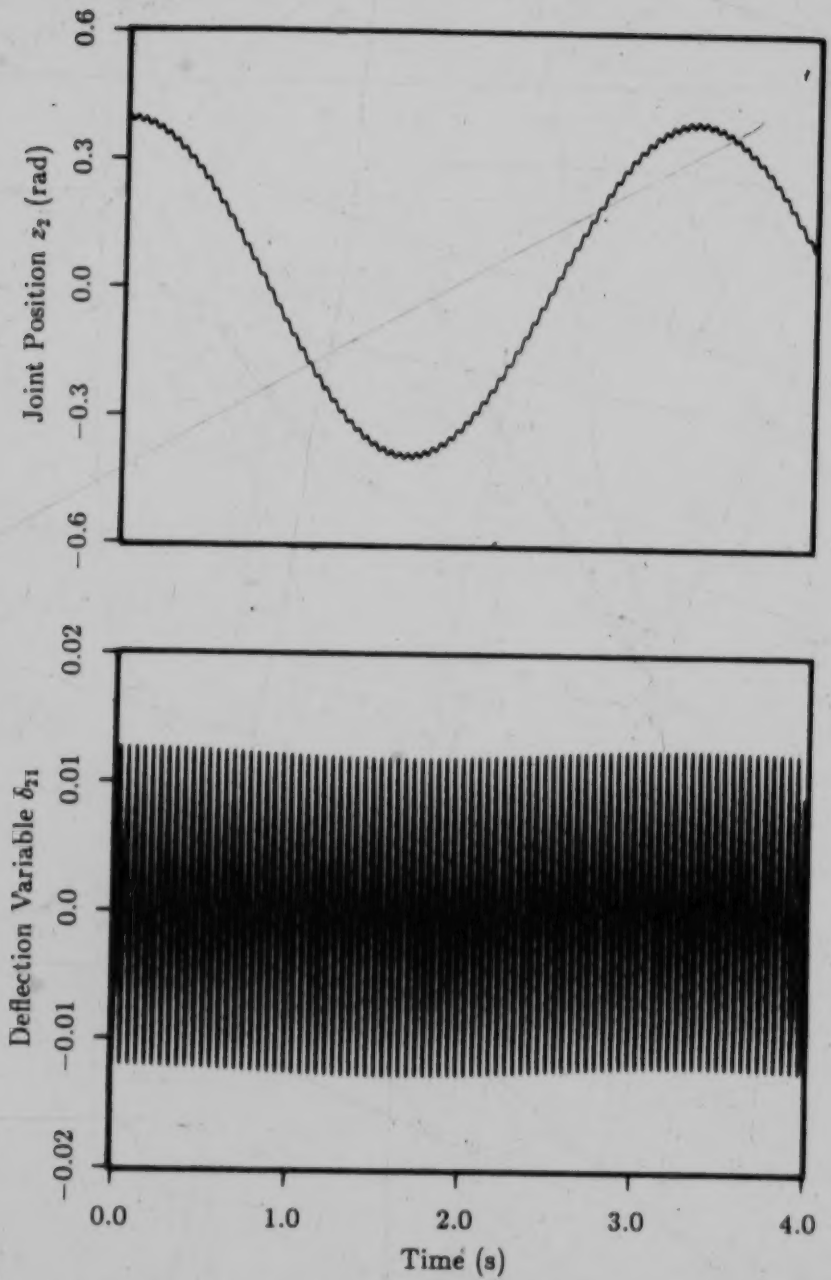

Figure 3.16: Response of a flexible link pendulum to an initial displacement and deflection, simulated using FLX with a 1 ms timestep

3.3.4 Single flexible link with a rigid joint mass

The fourth test case is a flexible link with a rigid joint mass (JMFLX), shown in Figure 3.17. A rigid joint mass is free to rotate about Z_0 and the flexible link is attached to the rigid joint mass. The same fundamental modeshape, used previously, is used to describe the flexure of the link. Gravitational force on the system is ignored. The equations of motion of the flexible link with a rigid joint mass are

$$\left\{m\left(R^2 + Ra + \frac{a^2}{3}\right) + mF\varsigma^2 + J\right\}\ddot{\phi} + m\left(RH + Ga\right)\ddot{\varsigma} + 2mF\varsigma\dot{\varsigma}\dot{\phi} = 0 \quad (3.17)$$

$$m\left(RH + Ga\right)\ddot{\varsigma} + mF\ddot{\varsigma} + \left(\frac{KEI}{4a^3} - mF\dot{\varsigma}^2\right)\varsigma = 0 \quad (3.18)$$

where

ϕ is the angular displacement of the undeformed link from its initial position,

ς is the modal coordinate describing the deflection of the link,

m is the mass of the beam,

R is the radius of the joint mass,

J is the polar moment of inertia of the joint mass, and

F, G, H, and K have the same definitions as in equations (3.11) and (3.12).

The FLX geometry used to model the system is shown in Figure 3.18. Link 1 is not free to rotate about $\hat{Z}_0$, joint 1 is locked. A rigid joint mass is free to rotate about $\hat{Z}_1$, and the flexible link 2 is attached to the rigid joint mass. Link 2 is given an initial deflection in the $\hat{X}_1\hat{Y}_1$ plane and released form rest.

The response of the system with no joint mass, simulated using JMFLX with an integration time step of 1 ms, to an initial deflection of $\varsigma = 0.012$ is shown

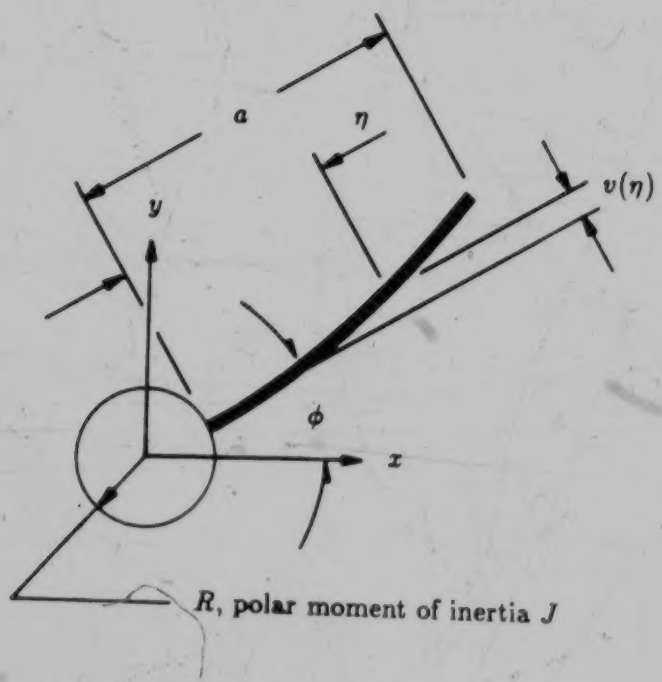

Figure 3.17: JMFLX geometry used to model a single flexible link with a rigid joint mass

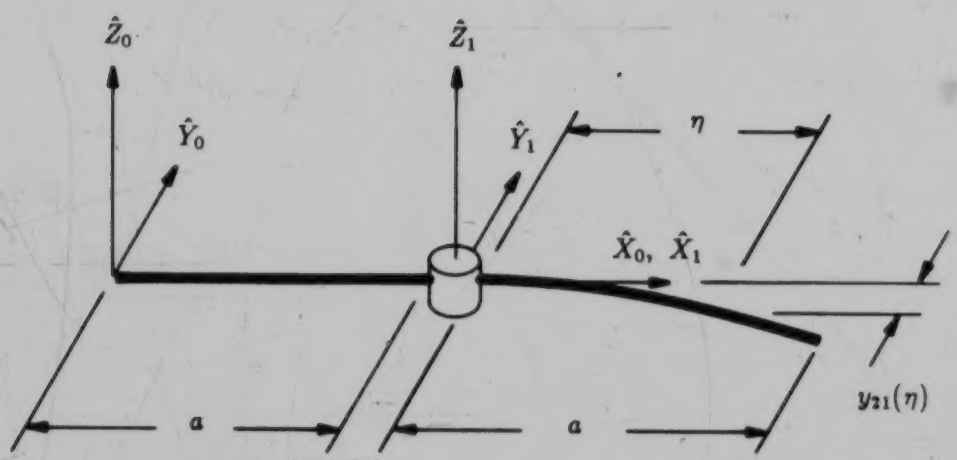

Figure 3.18: FLX geometry used to model the single flexible link with a rigid joint mass

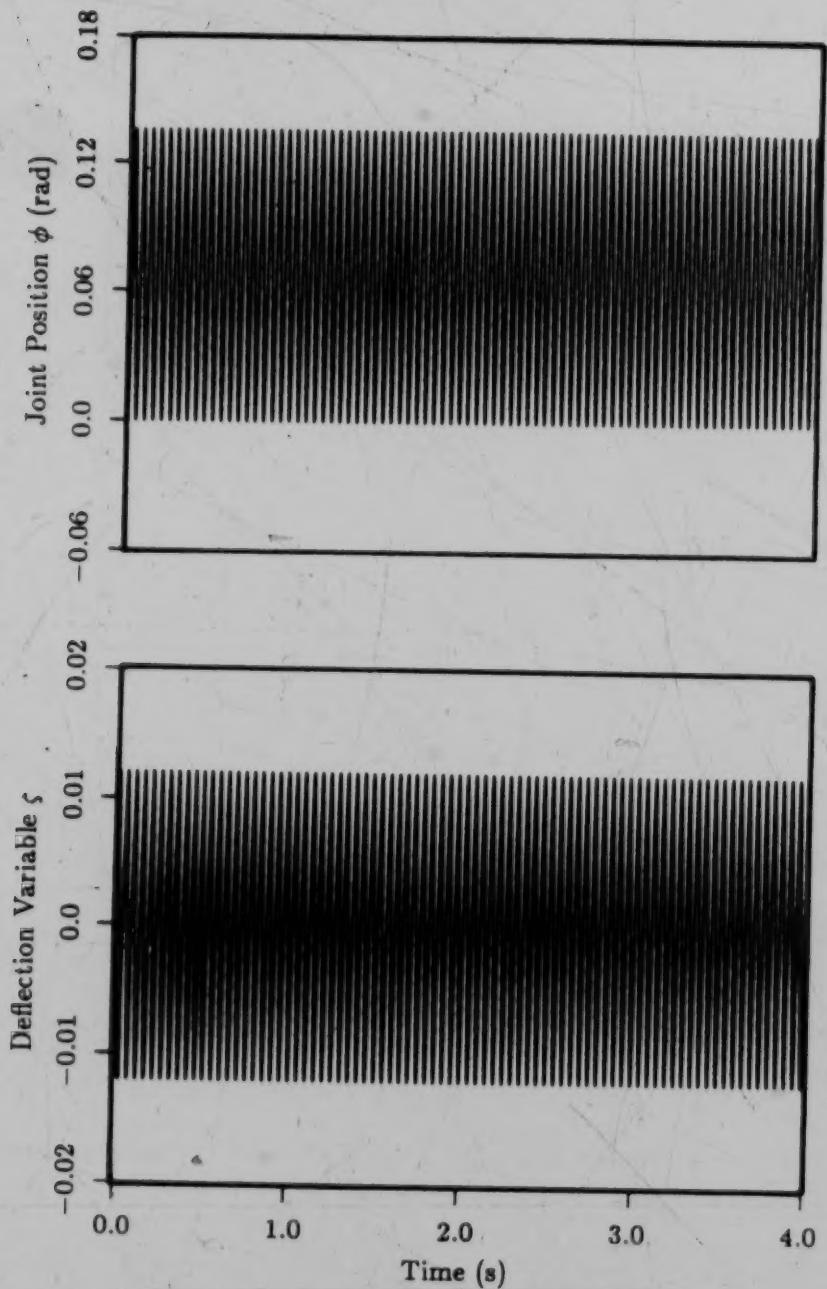

Figure 3.19: Response of a single flexible link with a rigid joint mass to an initial deflection of $\varsigma = 0.012$, simulated using JMFLX with a 1 ms timestep

in Figure 3.19. The period of the link flexural oscillation is 49.6 ms and the link vibration back drives the joint. The effect of increasing the joint inertia on the frequency of link flexural oscillation, simulated using both JMFLX and FLX, is shown in Figure 3.20. As the rotational inertia of the joint mass is increased, the amplitude of the joint oscillation is reduced and the period of the link flexural oscillation approaches the analytical value of 275 ms for the period of a fixed free beam, shown in Figure 3.20.

The most serious shortcoming in the model verification is the lack of an independent description of the link flexibility. Both FLX and the scalar Lagrangian test cases use the same truncated modal approximation to describe the flexure of the link. However, the asymptotic approach of the period of the link flexural oscillation to the analytical value for a fixed free beam as the joint inertia is increased is an indication that the implementation of link flexibility in FLX is correct. Unlike JMFLX, the FLX model has a mode of flexure in the $\hat{X}_1\hat{Z}_1$ plane, in the plane of the joint axis. The base of the link is fixed in relation to flexure in this plane. If the initial deflection is in the plane of the joint axis rather than in the plane normal to the joint axis, the link vibrates with the a period of 275 ms identical to the analytical value for a fixed free beam, as shown in Figure 3.21.

3.3.5 General move of a three link arm

The last test case is a general move of a three link manipulator with two flexible links. The motion in all the previous test cases has been planar, in this test case the motion is three dimensional. The geometry of the system is shown in Figure 3.22. The first link, which rotates about $\hat{Z}_0$, has zero length. The second link, which rotates about $\hat{Z}_1$, is one of the standard 4 m links with two modes of

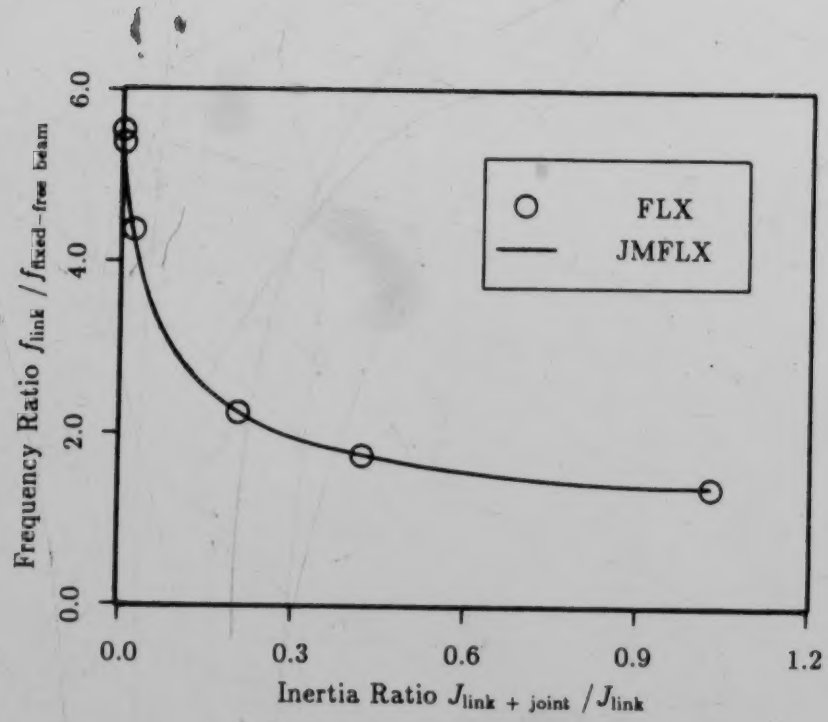

Figure 3.20: The effect of varying joint inertia on the frequency of link flexural oscillation

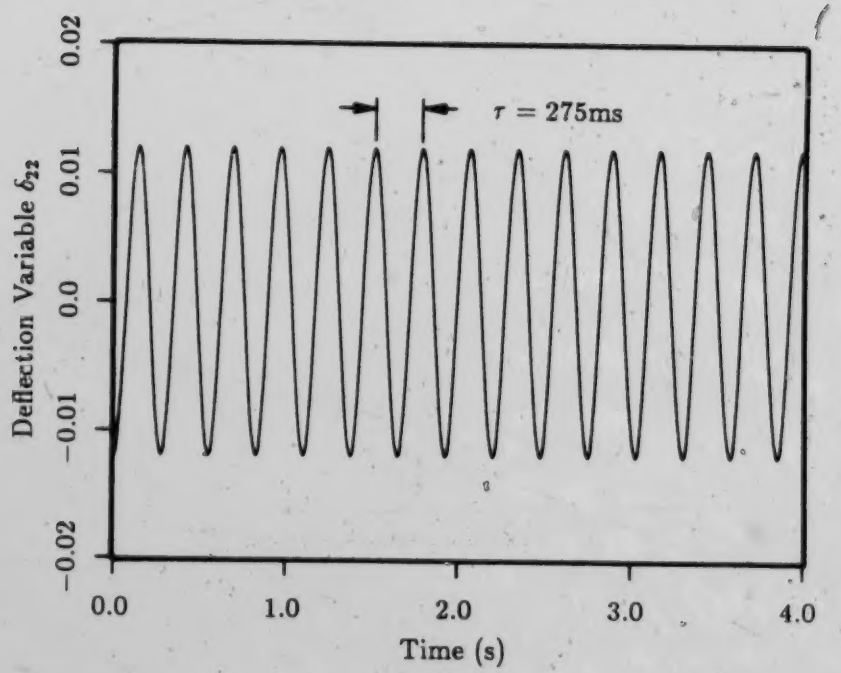

Figure 3.21: Response of the flexible link with a rigid joint mass to an initial deflection in the plane of the joint axis, simulated using FLX with a time step of 1 ms.

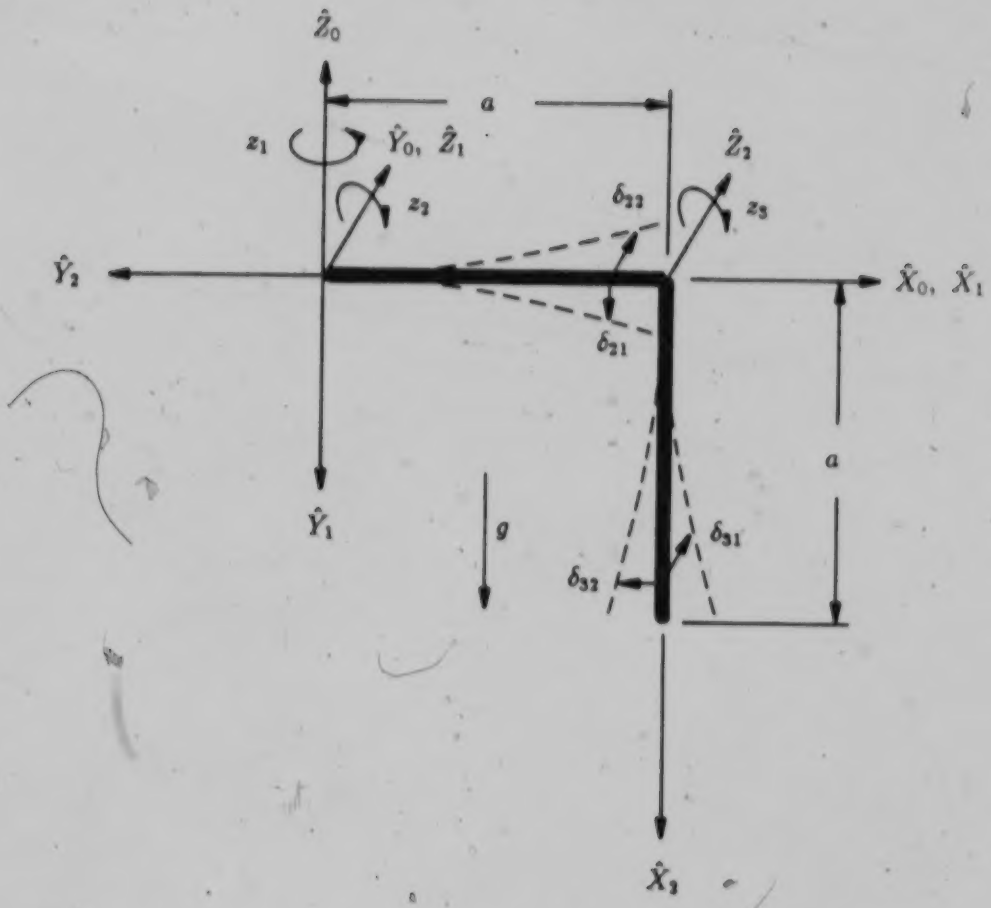

Figure 3.22: Three link manipulator

flexibility one in the local $\hat{X}\hat{Y}$ plane and one in the local $\hat{X}\hat{Z}$ plane. The third link, which rotates about $\hat{Z}_2$, is identical to the second. Thus, there are 7 degrees of freedom in total.

The command trajectory for the move is shown in Figure 3.23. The initial and final positions of the three link manipulator are shown in Figure 3.24. The joint torques are supplied by independent, fixed gain PID joint controllers. The input to the joint controllers is the joint position error, its time derivative, and integral. The output from the joint controllers are the joint torques. The joint controllers set the manipulator in motion and are then turned off at 0.5 s into the move. The response of the rigid and flexible link manipulators is shown in Figures 3.25, 3.26 and 3.27, respectively. When the joint controllers are switched off, there is no external energy input into the system. The solutions involving both the rigid and flexible models reflect this and are conservative, as shown in Figures 3.25 and 3.26.

The time history of the control joint torques are shown in Figure 3.28 for the case when the joint controls are not switched off at 0.5 s into the move. If the time history of the kinematic state of the model for this case is fed into the inverse dynamics calculation, the joint torques required to produce the motion are calculated. These joint torques, shown in Figure 3.29, are the same as the control joint torques input into the forward dynamics. Thus the FLX simulation demonstrates closure.

In this chapter, the implementation of the FLX simulation has been outlined and some attempts at model verification have been presented. The FLX simulation has been shown to produce correct, conservative solutions for planar and three dimensional rigid body motion in several test cases. This has also been demonstrated for flexible link motion in which the flexure of the link is described

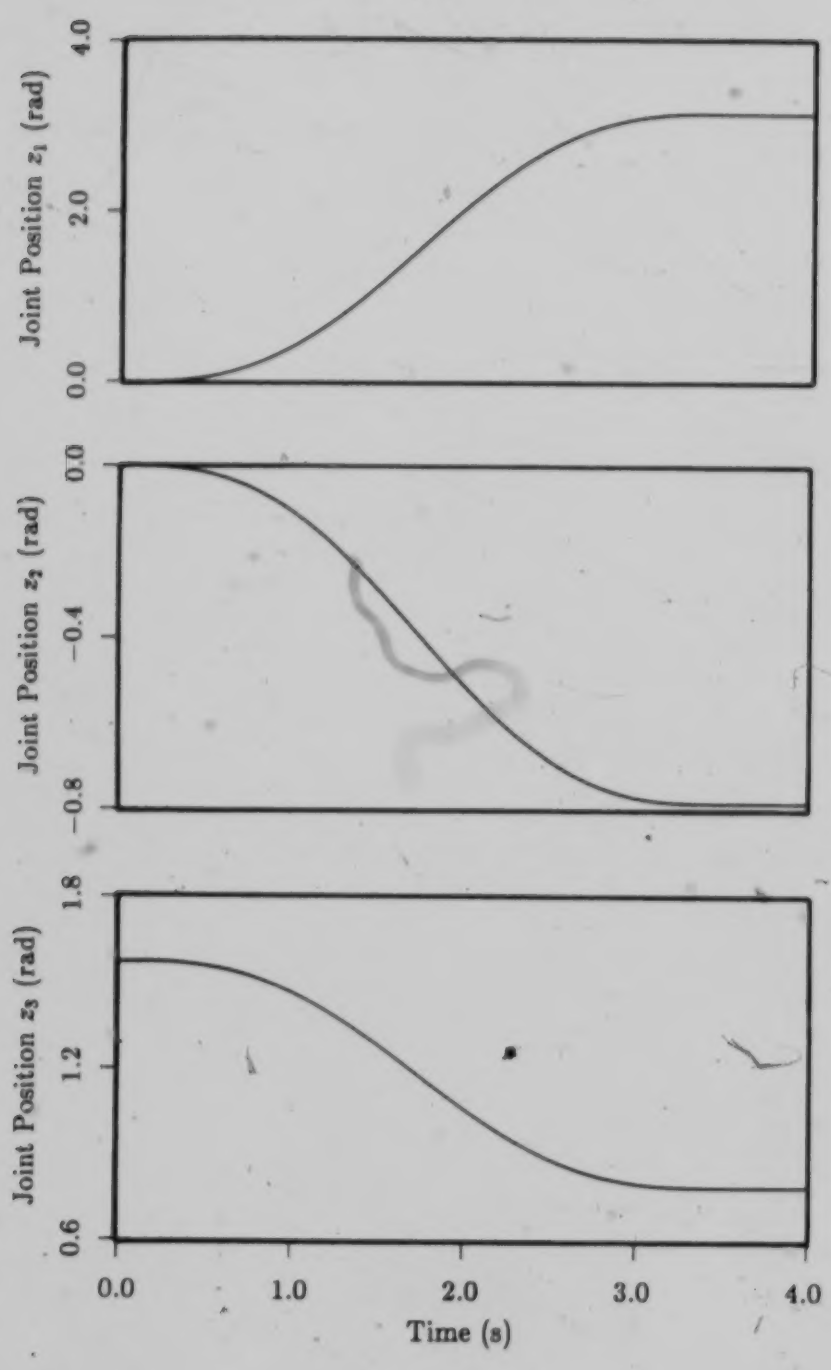

Figure 3.23: Command input to the three link manipulator

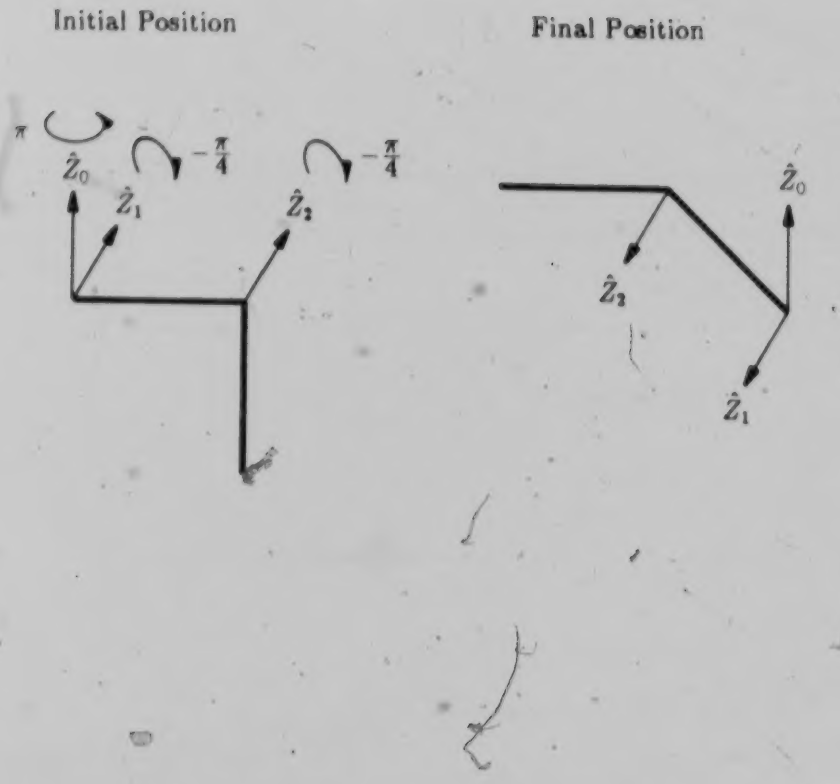

Figure 3.24: Initial and final positions of the three link manipulator

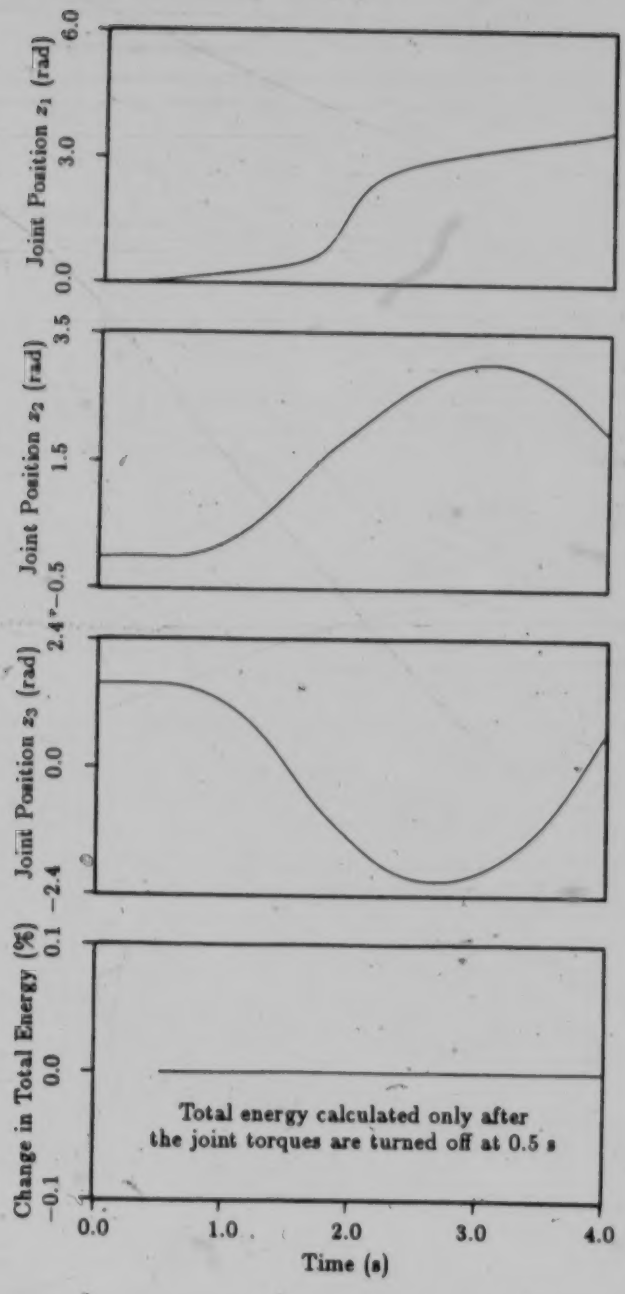

Figure 3.25: Response of the rigid link manipulator

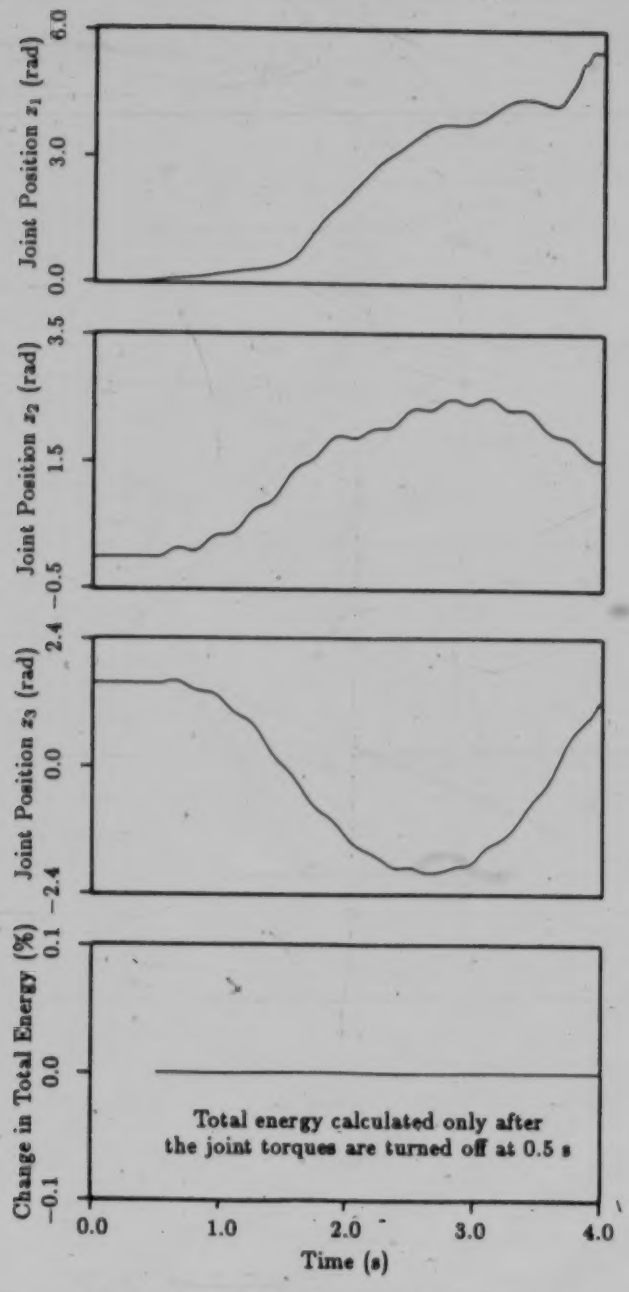

Figure 3.26: Response of the flexible link manipulator: joint positions and system energy

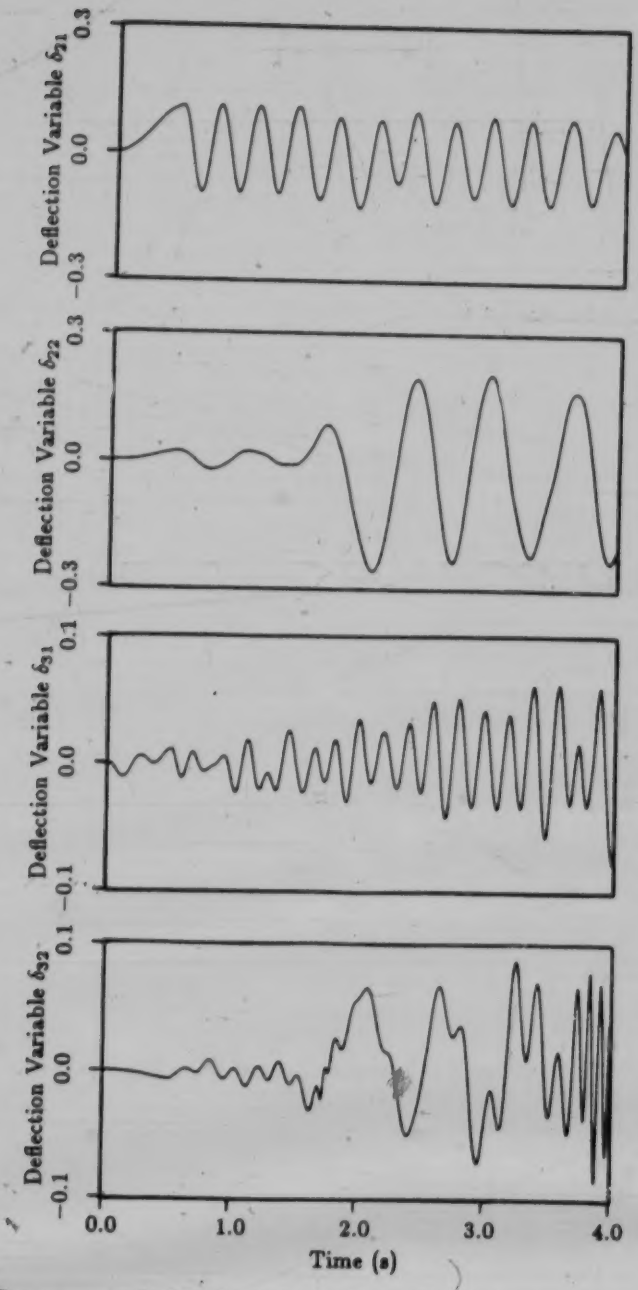

Figure 3.27: Response of the flexible link manipulator: generalized deflection variables

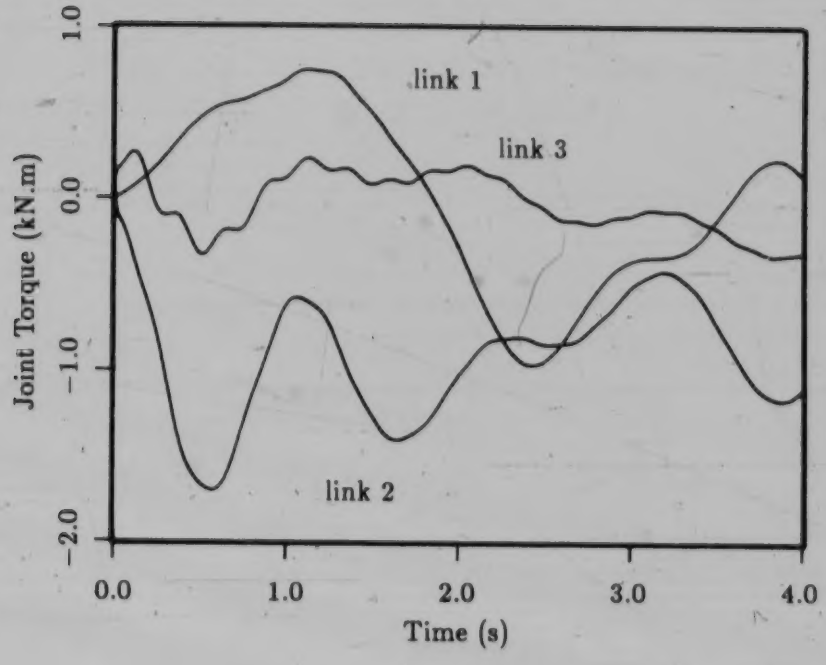

Figure 3.28: Time history of the control joint torques

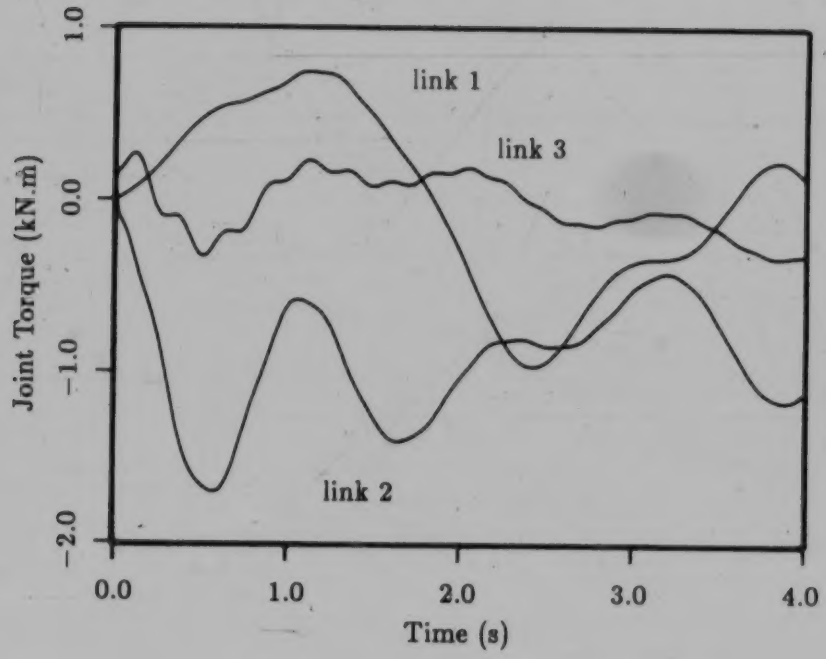

Figure 3.29: Time history of the joint torques from the inverse dynamics

by the fundamental mode of flexural vibration for a fixed free beam. The FLX simulation has been shown to produce the correct frequency for the oscillation of a rigid compound pendulum for small displacements. The FLX simulation has also been shown to produce the correct frequency for the fundamental mode of vibration of a fixed free beam. And finally, the FLX simulation demonstrates closure, control torques input into the forward dynamics are reproduced by inputting the kinematic state of the model into the inverse dynamics calculation. Although there is no conclusive proof of the correctness of FLX, it appears to produce the correct results for the test cases presented.

Chapter 4

Symbolic Generation of the Flexible Link Dynamics

In this chapter, symbolic generation of the flexible link dynamics equations based on Book's [27] algorithm, outlined in Chapter 2, is presented. The computational burden of the symbolically generated dynamics is compared to that of the numerical dynamics formulation.

Symbolic generation refers to the manipulation of symbols as opposed to numeric values. Symbolic generation offers the possibility of obtaining very efficient formulations of manipulator dynamics. Manipulations, such as matrix multiplications and vector cross products inherent in dynamics formulations, are performed only once, eliminating the use of loop counters and the overhead of subroutine calls to numerical matrix handling routines. Also, needless operations such as multiplication by 1 or 0 can be avoided and replaced by assignments.

Symbolic generation of the dynamics equations of rigid link manipulators is not new. Luh and Lin [35] developed an algorithm for symbolically generating the

manipulator dynamics using the Newton-Euler formulation. Vecchio et al. [36] derived the equations of motion, based on the Lagrangian formulation, using the symbol manipulation language REDUCE. Proponents of the use of Kane's dynamics equations have used manual symbolic generation to formulate the dynamics of multibody satellites and rigid link manipulators [1]. Nielen and Kane [37] have compared the efficiency of the Stanford manipulator inverse dynamics calculation produced by several symbolic generation schemes. The most efficient inverse dynamics calculation reported by these authors was Wampler's [37] formulation, an iterative/recursive method based on Kane's equations requiring only 468 multiplications and 332 additions. Recently, Toogood [23] has reported an inverse dynamics calculation for the Stanford manipulator requiring only 455 multiplications and 363 additions. This inverse dynamics calculation was derived symbolically using the DYNAM/CLEAR system, developed by Kermack [28] and Toogood [23], which generates the equations of motion based on Hollerbach's [17] 4×4 matrix Lagrangian formulation. If the inertia terms for the final three links plus the inertia terms for the first three links which are smaller by at least one order of magnitude than the largest term for that link are ignored, only 298 multiplications and 221 additions are required for the computation of the Stanford manipulator inverse dynamics [23].

DYNAM [23,28] is not a general purpose symbolic manipulation routine. Rather, it was written specifically to generate the inverse dynamics calculation based on Hollerbach's [17] 4×4 matrix Lagrangian formulation. DYNAM creates a FORTRAN subroutine consisting of a collection of dummy variables, Z### strings, where ### is a number. The Z### variables with the strings assigned to them constitute FORTRAN statements which encode the inverse dynamics calculation. All matrix and vector elements in Hollerbach's formulation are treated as sym-

bols and are stored in memory as ASCII strings. Multiplication is performed by concatenation of a FORTRAN multiplication operator "*" with the first string followed by concatenation with the second string. Addition is performed in a similar manner with an included addition "+" operator. Parentheses are used only to preserve algebraic hierarchy. Multiplication by 0, 1, or -1 is replaced by appropriate assignment. Addition of 0 is also not performed. The proliferation of strings of excessive length is a problem common to symbolic generation routines [1,28]. This problem is alleviated in DYNAM by replacing a string resulting from a symbolic operation, if it contains a plus "+" sign or two or more multiplication "*" signs, with a Z### variable in memory and assigning the original string to the Z### variable in the FORTRAN subroutine. For example, if a matrix or vector element contains a string

$$Z20 + Z13 * Z14$$

resulting from a symbolic operation, it would be replaced in memory by Z###, with the number ### being generated automatically, and the assignment

$$Z### = Z20 + Z13 * Z14$$

would be made in the FORTRAN subroutine.

Due to the especially sparse nature of some of the matrices, the application of symbolic generation to flexible link dynamics should result in a significant reduction in the computational load in comparison to the numerically coded dynamics. The C_i, C_{ij}, and C_{ikj} matrices, which describe the distribution of mass in a link, are particularly sparse. Expanding equation (2.29), the C_i matrices are seen to have only four non-zero elements,

$$
\begin{aligned}
C_i &= \frac{1}{2}\int_0^{a_i} \mu(\eta\ 0\ 0\ 1)^T(\eta\ 0\ 0\ 1)d\eta \\
&= \frac{1}{2}\begin{Bmatrix} \int_0^{a_i}\mu\eta^2 d\eta & 0\ 0 & \int_0^{a_i}\mu\eta\, d\eta \\ 0 & 0\ 0 & 0 \\ 0 & 0\ 0 & 0 \\ \int_0^{a_i}\mu\eta\, d\eta & 0\ 0 & \int_0^{a_i}\mu d\eta \end{Bmatrix}.
\end{aligned} \tag{4.1}
$$

Expanding equation (2.30), the C_{ij} matrices are also seen to be very sparse,

$$
\begin{aligned}
C_{ij} &= \frac{1}{2}\int_0^{a_i} \mu(\eta\ 0\ 0\ 1)^T(x_{ij}\ y_{ij}\ z_{ij}\ 0)d\eta \\
&= \frac{1}{2}\begin{Bmatrix} \int_0^{a_i}\mu\eta\, x_{ij}d\eta & \int_0^{a_i}\mu\eta\, y_{ij}d\eta & \int_0^{a_i}\mu\eta\, z_{ij}d\eta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \int_0^{a_i}\mu x_{ij}d\eta & \int_0^{a_i}\mu y_{ij}d\eta & \int_0^{a_i}\mu z_{ij}d\eta & 0 \end{Bmatrix}.
\end{aligned} \tag{4.2}
$$

Since axial deformation of a link was ignored in the derivation of FLX, the elements of the first column of the C_{ij} matrices are zero. If δ_{ij} is the time varying amplitude of the j_{th} mode of bending in the local $\hat{X}\hat{Y}$ plane, then only the second column of C_{ij} contains non-zero elements. And, if δ_{ij} is the time varying amplitude of the j_{th} mode of bending in the local $\hat{X}\hat{Z}$ plane, then only the third column of C_{ij} contains non-zero elements. Thus there are only two non-zero elements in C_{ij}. Expanding equation (2.31) gives

$$
\begin{aligned}
C_{ikj} &= \frac{1}{2}\int_0^{a_i} \mu(x_{ik}\ y_{ik}\ z_{ik}\ 0)^T(x_{ij}\ y_{ij}\ z_{ij}\ 0)d\eta \\
&= \frac{1}{2}\begin{Bmatrix} \int_0^{a_i}\mu x_{ij}x_{ik}d\eta & \int_0^{a_i}\mu y_{ij}x_{ik}d\eta & \int_0^{a_i}\mu z_{ij}x_{ik}d\eta & 0 \\ \int_0^{a_i}\mu x_{ij}y_{ik}d\eta & \int_0^{a_i}\mu y_{ij}y_{ik}d\eta & \int_0^{a_i}\mu z_{ij}y_{ik}d\eta & 0 \\ \int_0^{a_i}\mu x_{ij}z_{ik}d\eta & \int_0^{a_i}\mu y_{ij}z_{ik}d\eta & \int_0^{a_i}\mu z_{ij}z_{ik}d\eta & 0 \\ 0 & 0 & 0 & 0 \end{Bmatrix}.
\end{aligned} \tag{4.3}
$$

Since axial deformation of a link was ignored, $x_{ij} = 0$ for all i, j and the elements of the first row and column of C_{ikj} vanish. Due to the orthogonality of the modeshapes in bending, the remaining diagonal elements of C_{ikj} are zero for $j \neq k$. The modeshapes in bending in the local $\hat{X}\hat{Y}$ and $\hat{X}\hat{Z}$ planes are identical. However, for any mode j only one of y_{ij} or z_{ij} is non-zero. That is, δ_{ij} is the time varying amplitude of of a mode of flexural vibration in either the local $\hat{X}\hat{Y}$ or the $\hat{X}\hat{Z}$ plane, not both. Therefore, $\int_0^{a_i} \mu y_{ij} z_{ik} d\eta = 0$ for $j = k$. Thus the C_{ikj} have at most only two non-zero elements. The M_{ij} matrices are also particularly sparse,

$$M_{ij} = \begin{pmatrix} 0 & -\theta_{xij}(a_i) & \theta_{yij}(a_i) & x_{ij}(a_i) \\ \theta_{xij}(a_i) & 0 & -\theta_{xij}(a_i) & y_{ij}(a_i) \\ -\theta_{yij}(a_i) & \theta_{zij}(a_i) & 0 & z_{ij}(a_i) \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.4}$$

Again, since axial link deformation was ignored, $x_{ij} = 0$ for all i, j. If mode j represents bending in the local $\hat{X}\hat{Y}$ plane, then only θ_{xij} and y_{ij} are non-zero. Similarly, if mode j represents bending in the local $\hat{X}\hat{Z}$ plane, then only θ_{yij} and z_{ij} are non-zero. Thus the M_{ij} matrices contain at most only three non-zero elements.

A symbolic generation routine, SYMFLX, incorporating the features of DYNAM, has been written in C based on the flexible link dynamics presented in Chapter 2. The numerically coded FLX evaluates the forward dynamics, that is given the generalized positions, velocities, and the joint torques, the generalized accelerations are evaluated from equation (3.1)

$$\ddot{z} = I^{-1} R$$

and integrated to produce a time history of the kinematic state of the manipulator. SYMFLX follows the structure of FLX, performing the operations symbolically,

but does not perform the inversion of the inertia matrix or the time integration of equation (3.1). The inputs to the SYMFLX generated FORTRAN subroutine FORTSUB are the generalized positions $\boldsymbol{z}$, velocities $\dot{\boldsymbol{z}}$, and the applied joint torques $\boldsymbol{F}$. The mass and inertia properties of the links are incorporated into FORTSUB by SYMFLX. FORTSUB returns the inertia matrix $\boldsymbol{I}$ and the remaining dynamics vector $\boldsymbol{R}$. Using FORTSUB, the forward dynamics can be evaluated by performing the inertia matrix inversion and integrating equation (3.1) numerically as in FLX. The inverse dynamics can be evaluated by calling FORTSUB with the joint torques set to zero. On return, the value of the joint torques can be determined from the difference between the remaining dynamics vector calculated using equation (2.57)

$$\boldsymbol{I}\ddot{\boldsymbol{z}} = \boldsymbol{R}$$

and that returned by FORTSUB.

SYMFLX has been used to generate a FORTRAN subroutine that evaluates the inverse dynamics of the three link manipulator introduced in Chapter 3. The computational burden of the symbolically generated inverse dynamics calculation is compared to that of FLX in Table 4.1. For the purpose of this comparison, the added computation due to the inclusion of the rigid joint mass transformation, $\boldsymbol{S}_i$, has been ignored as this transformation could have been incorporated into the joint transformation, $\boldsymbol{A}_i$.

In the rigid link case, the numerically coded FLX requires almost as many operations as the Uicker-Kahn formulation, even though Book's [27] algorithm incorporates recursive schemes similar to those used by Hollerbach [17]. The majority of operations specific to the inclusion of link flexibility are avoided in the rigid link case through the use of logical constructs in FLX. Therefore, the large number of operations required by FLX in the rigid case must be due in part to

Number of Operations × = multiplication, + = addition				
Method	Operation	Number of Flexible Modes[a]		
		0	1	2
Uicker-Kahn	×	4998	—	—
	+	6484	—	—
Hollerbach	×	1898	—	—
(4 × 4)	+	1561	—	—
FLX	×	4637	10345	18117
	+	2715	8513	15223
SYMFLX[b]	×	638	3210	—
	+	300	1648	—
SYMFLX[c]	×	445	2659	—
	+	213	1313	—

[a]The number of modes in the local $\hat{X}\hat{Y}$ and $\hat{X}\hat{Z}$ of each flexible link in the model;

2 modes ⇒ 1 rigid link + 2 flexible links × (1 + 2 modes × 2 planes of bending)
= 11 degrees of freedom.

[b]The output of SYMFLX without any modification.

[c]The output of SYMFLX with unused Z### variables removed.

Table 4.1: Computational requirements for the inverse dynamics of a three link manipulator

increased number of matrix multiplications incurred as a result of the separation of the joint transformation, A_i, and the link transformation, L_i. The computational requirement of the symbolically generated inverse dynamics calculation, 445 multiplications and 213 additions, is less than $\frac{1}{4}$th of the number of operations required by Hollerbach's 4×4 matrix formulation and less then $\frac{1}{10}$th of the number of operations required by FLX. This is close to the computational requirement reported by Kermack [28], 450 multiplications and 243 additions, for a symbolically generated inverse dynamics calculation based on Hollerbach's [17] algorithm. It should be noted, however, that SYMFLX removes unused Z### strings, whereas Kermack's routine did not.

In the flexible link case, with one bending mode in the local $\hat{X}\hat{Y}$ and one in the local $\hat{X}\hat{Z}$ plane of each of the two flexible links, FLX requires 10345 multiplications and 8513 additions to compute the inverse dynamics. In comparsion, SYMFLX, generated 1870 Z### strings, 324 of which were unused and deleted. The remaining 1546 Z### strings contained 2659 multiplications and 1313 additions required to compute the inverse dynamics, roughly $\frac{1}{4}$th of the computational burden of FLX.

Estimating the increase in computational efficiency of the symbolically generated inverse dynamics calculation over the numerical FLX routine by counting the number of floating point operations is somewhat misleading. The FLX inverse dynamics calculation involves a large number of subroutine calls to matrix handling routines whereas the SYMFLX inverse dynamics calculation only contains one subroutine call. The FLX routine also employs loop counters, logical testing, pointers, and array indexing, all of which are absent from the SYMFLX inverse dynamics calculation. Instead a more direct comparison of execution speed on the same hardware and operating system is a better indication of relative efficiency.

Execution Time[a] (ms)		
Method	Number of Flexible Modes[b]	
	0	1
FLX	233	403
SYMFLX[c]	4	78

[a] Average times for a call to the dynamics subroutine on a MC68020 (20 MHz operation) coprocessor board ©Definicon System Inc.

[b] The number of modes in the local $\hat{X}\hat{Y}$ and $\hat{X}\hat{Z}$ of each flexible link in the model;

2 modes $\Rightarrow$ 1 rigid link + 2 flexible links $\times$ (1 + 2 modes $\times$ 2 planes of bending)
= 11 degrees of freedom.

[c] The output of SYMFLX with unused Z### variables removed.

Table 4.2: Execution times for the inverse dynamics calculation.

The execution times for the three link manipulator are shown in Table 4.2. In the rigid link case, the symbolic inverse dynamics calculation executes approximately 58 times faster than the FLX calculation. However, in the flexible link case, the symbolic inverse dynamics calculation is only about 5 times faster than FLX.

No simplification of the SYMFLX dynamics subroutine, other than removal of the unused Z### variables, was performed. An examination of the SYMFLX dynamics subroutine shows the presence of a number of repeated strings. For example, the SYMFLX dynamics subroutine for the single flexible mode case con-

tains 48 occurrences of "$C0 * S1$" and "$S0 * S1$" and 37 occurrences of "$C0 * C1$" and "$C0 * C1$" for a total of 166 duplicated multiplications. A repeated string could be replaced by assignment to a Z### variable on the first encounter and by that Z### variable on subsequent occurrences in the dynamics subroutine. This has been done by Toogood [23] in the CLEAR subroutine which removes unused Z### strings, replaces repeated strings, removes common factors from expressions, and simplifies some trigonometric identities. Using CLEAR on the DYNAM generated Stanford manipulator inverse dynamics, retaining all the inertia terms, Toogood [23] has reduced the number of operations required by more than 30% from 799 multiplications and 516 additions to 525 multiplications and 368 additions. Similar gains should be possible with the SYMFLX generated flexible link inverse dynamics.

The symbolic generation of the inverse dynamics of the three link manipulator with two bending modes in the local $\hat{X}\hat{Y}$ and two in the local $\hat{X}\hat{Z}$ plane of each of the two flexible links was not performed due to excessive memory requirements. All the symbolic arrays in SYMFLX are declared external, this is equivalent to common block storage in FORTRAN. Using the minimum permissible length of the symbolic strings of 50 characters for SYMFLX running the single mode problem, the storage required for the two mode problem exceeded the available 640 kilobytes. Work is in progress to restructure SYMFLX, using dynamic memory allocation for the symbolic strings. This should significantly increase the size of model that can be handled, since most of the symbolic strings are much shorter than the 50 character limit.

Chapter 5

Summary and Conclusions

Lightweight flexible manipulators hold the promise of increased speed of operation, longer reach, reduced power consumption, and lowered mounting strength and rigidity requirement. However, flexural and torsional oscillations may be excited by the control torques, degrading endpoint position, tracking performance, and cycle time of flexible manipulators. Unless these oscillations can be controlled, the advantages of lightweight, flexible manipulators will not be realized in practice.

Fast and accurate calculation of the dynamics is required in design and operation of robotic manipulators. Anticipated joint torques, based on an input trajectory, can be calculated online and fed to the joints for improved trajectory following. These calculations must be carried out at the servo rate, typically at least 60 times a second [4]. The computational burden of flexible link dynamics formulations has not been reported in the literature. The possible improvements in computational efficiency through the use of symbolic generation of the flexible link dynamics have not been demonstrated. To address these concerns a flexible link simulator based on a numerical algorithm of the equations of motion was

developed; and its computational load was compared to symbolically generated dynamics routines.

The derivation of the equations of motion, based on Book's algolrithm [27] was presented in Chapter 2. Book's algorithm uses a 4×4 matrix Lagrangian formulation to derive the equations of motion for a flexible link manipulator. Recursive schemes similar to those used by Hollerbach [17] are employed, but are more complex due to the separation of the joint transformation from the link transformation.

In Chapter 3, the implementation of Book's algorithm in the simulator FLX was presented and attempts at verifying its correctness were described. The simulator only models revolute joints, not prismatic ones. The modeshapes of a fixed free beam, eigenfunctions of the Bernoulli-Euler flexure equation, were used to describe the flexure of a link using the method of assumed modes.

Without a physical model with which to compare, absolute verification of the numerical simulator is difficult. However, some checks of the correctness of the simulator have been made. FLX has been shown to produce solutions that agree with simpler rigid body models involving motion in a plane, such as the single and double link compound pendula. FLX has also been shown to produce conservative solutions for the flexible link compound pendulum provided that the integration time step used in the fourth order Runge-Kutta scheme is not larger than approximately $\frac{1}{50}$th of the period of the link flexural oscillations. FLX has been shown to produce conservative solutions of the equations of motion of a rigid and a flexible three joint, two link manipulator in the absence of external forcing. And FLX has also been shown to demonstrate closure: control torques input into the forward dynamics are reproduced by inputting the kinematic state of the model into the inverse dyanmics calculation. However, no independent check of the flexible link

dynamics has been made. The same mode summation used to describe flexure in FLX was used in the simpler models.

In Chapter 4, the symbolic generation of the flexible link dynamics was presented. Symbolic generation of the flexible link dynamics significantly reduced the computational burden of the inverse dynamics calculation. For a two flexible link manipulator with three rigid body degrees of freedom and four bending modes, seven degrees of freedom in total, symbolic generation of the equations of motion reduced the number of floating point operations required from 10345 multiplications and 8513 additions to 2659 multiplications and 1313 additions, resulting in an approximately fivefold increase in execution speed. However, the increase in execution speed was not as dramatic as that for the rigid link manipulator of the same geometry for which the symbolic generation of the symbolic dynamics executed approximately 58 times faster than FLX. These increases in computational efficiency were obtained without any sophisticated editing of the symbolic dynamics subroutine, only unused strings were removed. There were a large number of repeated strings in the symbolic dynamics subroutine showing that further simplification of the symbolic dynamics will result in improved computational efficiency.

Since most of the symbolic strings generated are much shorter than the longest, static storage of the symbolic matrices, composed of character arrays of sufficient length to store the longest string, is wasteful of memory. It is recommended that dynamic memory allocation be used in the storage of the symbolic matrices rather than static storage, in order to increase the size of the problem that can be handled.

References

[1] Kane, T.R. and Levinson, D.A. 1983 The use of Kane's dynamical equations in robotics. *International Journal of Robotics Research* **2**(3):3-21.

[2] Luh, J.Y.S., Walker, M.W., and Paul, R.P.C. 1980 On-line computational scheme for mechanical manipulators. *Trans. ASME J. Dyn. Syst. Measurement Contr.* **102**:69-76.

[3] Luh, J.Y.S. 1983 An anatomy of industrial robots and their controls. *IEEE Trans. Automatic Control* **28**(2):133-153.

[4] Paul, R.P.C. 1981 **Robot Manipulators: Mathematics, Programming, and Control**, The MIT Press, Cambridge, Massachusetts.

[5] Book, W.J. and Majette, M. 1983 Controller design for flexible, distributed parameter mechanical arms via combined state space and frequency domain techniques. *Trans. ASME J. Dyn. Syst. Measurement Contr.* **105**:245-254.

[6] Usoro, P.B., Nadira, R., and Mahil, S.S. 1984 Control of lightweight flexible manipulators: A feasibility study. *Proc. 1984 American Control Conference*, San Diego, Calif. **2**:1209-1216.

[7] Book,W.J., Maïssa-Neto, O., and Whitney, D.E. 1975 Feedback control of

two beam, two joint systems with distributed flexibility. *Trans. ASME J. Dyn. Syst. Measurement Contr.* **97**(4):424-431.

[8] Book, W.J. 1979 Analysis of massless elastic chains with servo controlled joints. *Trans. ASME J. Dyn. Syst. Measurement Contr.* **101**(3):187-192.

[9] Chalhoub, N.G. and Ulsoy, A.G. 1984 Dynamic simulation of a flexible robot arm and controller. *Proc. 1984 American Control Conference*, San Diego, Calif. **2**:631-637.

[10] Singh, S.N. and Schy, A.A. 1985 Decomposition and state variable feedback control of elastic robotic systems. *Proc. 1985 American Control Conference*, Boston, Massachusetts, **2**:375-380.

[11] Hastings, G.G. and Book, W.J. 1985 Experiments in optimal control of a flexible arm. *Proc. 1985 American Control Conference*, Boston, Massachusetts, **2**:728-729.

[12] Sunada, W. and Dubowsky, S. 1983 On the dynamic analysis and behaviour of industrial robotic manipulators with elastic members. *Trans. ASME J. Mech., Trans., and Automation in Design* **105**:42-51.

[13] Pieper,D.L. 1968
cf. Brady et al. [22].

[14] Uicker, J.J. 1965 On the dynamic analysis of spatial linkages using 4 by 4 matrices. Ph.D. Thesis. Dept. of Mechanical Engineering and Astronautical Sciences, Northwestern University.
cf. Brady et al. [22].

[15] Kahn, M.E. and Roth B. 1969 The near-minimum-time control of open-loop articulated kinematic chains. *Stanford Artificial Intelligence Memo* **106**.

[16] Waters, R.C. 1979 Mechanical arm control, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, AIM 549
cf. Brady et al. [22].

[17] Hollerbach, J.M. 1980 A recursive formulation of Lagrangian manipulator dynamics and a comparative study of dynamics formulation complexity. *IEEE Trans. Systems, Man, and Cybernetics* **10**(11):730-736.

[18] Hooker, W. and Margulies, G. 1965 The dynamical attitude for an n-body satellite. *J. Astronautical Sciences* **12** 1965.

[19] Stepanenko, Y. and Vukobratovic, M. 1976 Dynamics of articulated open-chain active mechanisms. *Mathematical Biosciences* **28**:137-170.

[20] Orin, D.E., McGhee, R.B., Vukobratovic, M., and Hortoch, G. 1979 Kinematic and kinetic analysis of open-chain linkages utilizing Newton-Euler methods. *Mathematical Biosciences* **43**:107-130.

[21] Luh, J.Y.S., Walker, M.W., and Paul, R.P.C. 1980 Resolved acceleration control of mechanical manipulators. *IEEE Trans. Automatic Control* **25**:468-474.

[22] Brady, M., Hollerbach, J., Johnson, T., Lozano-Perez, T., and Mason, M. eds. 1982 **Robot Motion: Planning and Control**, The MIT Press, Cambridge, Massachusetts.

[23] Toogood, R.W. 1987 Symbolic generation of robot dynamics equations,

Part I: The DYNAM/CLEAR system. ACMIR TR 87-04, University of Alberta.

[24] Kelly, F.A. and Huston, R.L. 1981 Modeling of flexibility effects in robot arms *Proc. 1981 Joint Automatic Control Conference*, Green Valley, Arizona, Paper WP-2C.

[25] Singh, R.P. and Likins, P.W. 1983 Manipulator interactive design with interconnected flexible elements. *Proc. 1983 Joint Automatic Control Conference*, Paper TA6:505-512.

[26] Sunada, W. and Dubowsky, S. 1981 The application of finite element methods to the dynamic analysis of flexible spatial and co-planar linkage systems. *Trans. ASME J. Mechanical Design* **103**:643-651.

[27] Book, W.J. 1984 Recursive Lagrangian dynamics of flexible manipulator arms. *International Journal of Robotics Research* **3**(3):87-101.

[28] Kermack, I. 1986 The Effect of Misalignments on Manipulator Performance. M.Sc. Thesis, Dept. of Mechanical Engineering, University of Alberta.

[29] Kahn, M.E. and Roth B. 1971 The near-minimum-time control of open-loop articulated kinematic chains. *Trans. ASME J. Dyn. Syst. Measurement Contr.* **93**(3):164-172.

[30] Timoshenko, S. and Gere, J. 1972 **Mechanics of Materials**, Litton Educational Publishing.

[31] McCallion, H. 1973 **Vibration of Linear Mechanical Systems**, John Wiley and Sons.

[32] Denavit, J. and Hartenberg, R. S. 1955 A kinematic notation for lower-pair mechanisms. *Trans. ASME Journal of Applied Mechanics* June,1955:215-221.

[33] Timoshenko, S., Young, D.H., and Weaver, W. Jr. 1974 **Vibration Problems in Engineering** 4th ed., John Wiley and Sons.

[34] Goldstein, H. 1980 **Classical Mechanics**, Addison-Wesley, Reading, Massachusetts.

[35] Luh, J.Y.S. and Lin, C.S. 1981 Automatic generation of dynamic equations for mechanical manipulators. *Proc. 1981 Joint Automatic Control Conference*, Charlottesville, Virginia.

[36] Vecchio, L., Nicosia, S., Nicola, F., and Lentini, D. 1980 Automatic generation of dynamical models of manipulators. *Proc. 10th International Symposium on Industrial Robotics*, Milan, Italy, Mar. 1980:293-301.

[37] Nielan, P.E. and Kane, T.R. 1986 Symbolic generation of efficient simulation/control routines for multibody systems. in **Dynamics of Multibody Systems**, IUTAM/IFToMM Symposium, Udine, Italy. (G. Bianchi and W. Schiehlen, eds.) Springer-Verlag, Berlin.
cf. Toogood [23]

Appendix A

Denavit-Hartenberg Convention for Homogeneous Link Transformation Matrices

To specify the spatial relationship of a rigid body with respect to a fixed reference frame, it is necessary to specify the position of the origin of a coordinate frame fixed in the body and the orientation of the body fixed frame with respect to the reference frame. In general, six parameters are required, three to describe the position of the body fixed frame and three to describe its orientation with respect to the reference frame [34]. Consider a coordinate frame i embedded in link i at its distal end and a coordinate frame $i-1$ embedded in link $i-1$ at its distal end. In general six parameters are needed to describe the position and orientation of frame i relative to frame $i-1$. The Denavit-Hartenberg [32] convention for describing rigid link kinematics places restrictions on the allowable relative position and orientation of frames $i-1$ and i, reducing the number of

parameters needed to describe the position and orientation of frame i relative to frame $i-1$ to four. The parameters used are the joint angle θ_i, the link length a_i, the twist of the link α_i, and the link offset d_i. In the case of a revolute joint, the joint angle is the controlled variable; for a prismatic joint, the controlled variable is the link length.

The Denavit-Hartenberg convention for a revolute joint is shown in Figure A.1. With this convention, the origin of the coordinate system of link i is located at the intersection of the common normal between the axes of joints i and $i+1$ and axis of joint $i+1$. If the joint axes intersect, the origin of link i is at the point of intersection. If the joint axes are parallel, the origin of link i is chosen such that the offset of the next link whose coordinate frame is defined will be zero. The Z_i axis is colinear with the axis of joint $i+1$. The X_i axis is colinear with the common normal existing between joints i and $i+1$. If the joint axes intersect, the direction of the X_i axis is parallel or antiparallel to the cross product $Z_{i-1} \times Z_i$. The rotation of joint i is zero when X_{i-1} and X_i are parallel and have the same direction.

The Denavit-Hartenberg link transformation matrix A_i can be thought of as the product of the following elementary transformations:

1. rotation about Z_{i-1} equal to the joint angle θ_i ;
2. translation along the Z_{i-1} axis a length equal to the offset of the link d_i;
3. translation along the rotated X_{i-1} (now colinear with X_i) a distance equal to the length a_i of the link; and
4. rotation about X_i equal to the twist α_i of the link.

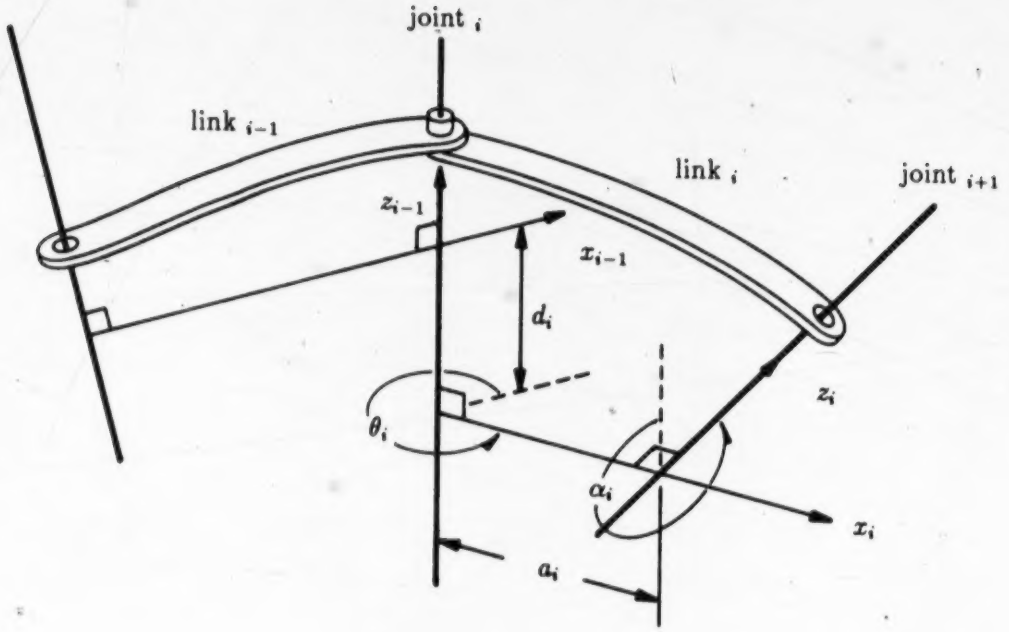

Figure A.1: The Denavit-Hartenberg convention for revolute joints

Forming the product of these elementary transformations results in the Denavit-Hartenberg link transformation matrix A_i for a rigid link i, relating the position and orientation of the coordinate frame of link i to the coordinate frame of link $i-1$,

$$A_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

To specify the position and orientation in space of the distal end of a link it is sufficient to specify the homogeneous transformation describing the relative position and orientation of the coordinate frame embedded at the distal end of the link, frame i, with respect to the coordinate frame whose origin is located at the base of the link, frame $i-1$. Starting at the base of a serial chain of links, the position of each link can then be specified relative to its predecessor in the chain. And the position of link i relative to the base frame is just the cumulative product of the link transformations starting from the base of the serial chain.